

# **From container terminals to bulk ports: models and algorithms for integrated planning and robust scheduling**

THÈSE N° 6149 (2014)

PRÉSENTÉE LE 17 AVRIL 2014

À LA FACULTÉ DE L'ENVIRONNEMENT NATUREL, ARCHITECTURAL ET CONSTRUIT  
LABORATOIRE TRANSPORT ET MOBILITÉ  
PROGRAMME DOCTORAL EN GÉNIE CIVIL ET ENVIRONNEMENT

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

**Nitish UMANG**

acceptée sur proposition du jury:

Prof. R. Bernier-Latmani, présidente du jury  
Prof. M. Bierlaire, directeur de thèse  
Prof. J. F. Cordeau, rapporteur  
Prof. A. L. Erera, rapporteur  
Prof. D. Kuhn, rapporteur



ÉCOLE POLYTECHNIQUE  
FÉDÉRALE DE LAUSANNE

Suisse  
2014



# Acknowledgments

First and foremost, I would like to thank my advisor Prof. Michel Bierlaire, who apart from being a great researcher and mentor, is also one of the kindest and most considerate people I have come across. I cannot thank him enough for his constant support and encouragement, and consider myself extremely fortunate to have him as my advisor.

I would like to express my gratitude to Prof. Alan Erera who hosted me at Georgia Tech. I have benefited a lot from his guidance and feedback. I would also like to thank other members of my thesis committee, Prof. Jean-Francois Cordeau and Prof. Daniel Kuhn for their valuable comments and suggestions, and Prof. Rizlan Bernier-Latmani who served as the president of the committee. I joined the TRANSP-OR lab as an intern in 2010. I shared office with Lissy La Paix for the first few months and I have some fond memories of those times. For most part of my Ph.D. I shared office with Bilal Farooq, and I would like to thank him for all the great conversations and good times.

I am deeply grateful to the ever smiling Marianne Ruegg for her assistance in all matters concerning life in Switzerland and travel abroad. I also wish to thank Anne for her kind and patient support with all the technical issues.

I worked with Ilaria Vacca on my first paper and I think I gained a lot from her expertise in port optimization. She is also one of the most caring and friendly people I know. I have also collaborated on research with Tomas and Jianghang, and have really enjoyed our discussions and working with them.

I would like to extend my heartfelt thanks to all the former and current members of TRANSP-OR- apart from those mentioned above, Aurelie, Bilge, Prem, Antonin, Thomas, Javier, Ricardo, Jingmin, Dimitrios, Eva, Amanda, Flurin, Anna, Marija, Stefan, Riccardo and Iliya for creating such a warm and friendly atmosphere at work.

My journey over these last few years would have been a lot more arduous had it not been for my friends in India and US. A big thanks to all of them.

Lastly and most importantly, I could not have come this far without the unconditional support of my family - my parents, my sister, my brother-in-law and my niece. They are my biggest strength and I dedicate this thesis to them.



# Abstract

In the operations research (OR) literature on port operations planning, there are a significant number of studies addressing decision problems in the context of container terminal management. Bulk terminals on the other hand, have been largely ignored. In this thesis, we study some of the key decision problems such as the berth allocation problem and the yard assignment problem in the bulk context.

The berth allocation problem (BAP) in bulk ports differs from that in container terminals, primarily because it is necessary to explicitly account for the cargo type on the vessel and the locations of the fixed equipment facilities such as conveyors and pipelines that are installed at only certain sections along the quay. We develop exact and heuristic algorithms to solve the BAP in bulk ports. The results based on instances inspired from real bulk port data look promising and suggest that the proposed methods can be successfully used to improve the operational efficiency of berth scheduling in bulk ports.

The BAP model is later extended and solved in integration with the yard assignment problem, that is, the problem of assigning different cargo types to specific locations in the yard. We propose a sophisticated exact solution algorithm based on the branch-and-price framework to solve the combined problem of berth allocation and yard assignment, which in all the previous studies related to container terminals has been solved using metaheuristics. Computational results based on real bulk port data suggest that the proposed algorithm can be successfully used to solve realistic sized instances in a computational time that is reasonable enough for the algorithm to be actually implemented and put into practice at the port.

Another key challenge in port operations planning is to address the enormous amount of uncertainty on account of factors such as weather conditions, mechanical problems and labor inefficiency among others. A stochastic disturbance can possibly render the planned schedules infeasible, thus incurring high costs to the port. In the current literature, there are very few studies related to handling uncertainty in port operations. In this thesis, we propose innovative models and solution techniques to handle uncertainty in scheduling, based on

## Acknowledgments

---

both proactive and reaction-based approaches.

We solve the berth allocation problem on a rolling planning horizon for a given planned baseline schedule and uncertainty in the arrival times and handling times of the vessels. The schedule is updated in response to disruptions as the actual arrival and handling times of the vessels are revealed in real-time. We propose recovery algorithms based on re-optimization and a smart greedy approach to reassign and reschedule the vessels, with the objective to minimize the total realized costs of the modified berthing schedule. The uncertainty in the yet-to-be-revealed part of the information is modeled by making appropriate assumptions about the probability distributions of the uncertain parameters derived from past data. The results suggest that our proposed methodology can significantly reduce the incurred costs as compared to the ongoing practice of reassigning vessels at the port.

To demonstrate the complexity in handling uncertainty in a proactive manner, we do a theoretical analysis of the most basic scheduling problem in the literature, that is, the single machine scheduling problem. In the context of port operations planning, the problem is analogous to the discrete berth allocation problem with a single berth that can handle at most one vessel at a given time. In all the previous studies on robust scheduling, the uncertainty in the release times of the jobs is largely ignored. We consider uncertainty in both the release times and the processing times of the jobs, discuss important properties of robust scheduling in the context of the single machine scheduling problem, and propose heuristics to generate robust schedules.

To summarize, this thesis makes significant fundamental contributions in both methodology and applications of OR. On the application side, we study the decision problems arising in bulk terminals, and propose innovative methods to solve these problems. On the methodological front, we address the problem of handling uncertainty in transportation and logistics systems planning in specific, and scheduling problems in general. Finally, the research presented in this thesis opens up several interesting and challenging possibilities for future research, particularly in the field of port operations planning.

**Keywords:** maritime logistics, bulk ports, container terminals, scheduling, mixed integer programming, metaheuristics, column generation, decision making under uncertainty.

# Résumé

Dans la littérature en recherche opérationnelle (RO) sur la planification des opérations portuaires, un nombre important d'études se focalisent sur les problèmes de décision dans le contexte de la gestion des terminaux à conteneurs. Or les terminaux de vrac sont largement ignorés. Dans cette thèse, nous étudions des problèmes de décision clés, tels le problème d'attribution de postes d'amarrage et le problème d'attribution de zones de stockage dans le contexte de ports vraquiers.

Le problème d'attribution de postes d'amarrage (APA) dans les ports vraquiers diffère de celui des terminaux à conteneurs, essentiellement parce qu'il est nécessaire de tenir compte explicitement du type de chargement du navire et de la localisation de l'équipement fixe, tel les convoyeurs et pipelines qui sont installés seulement sur certaines sections du quai. Nous développons des algorithmes exacts et heuristiques pour résoudre le APA dans les ports vraquiers. Les résultats basés sur des cas inspirés de données réelles d'un port sont prometteuses et suggèrent que les méthodes proposées peuvent être utilisées avec succès pour améliorer l'efficacité opérationnelle de la planification de l'attribution de postes d'amarrage dans les ports vraquiers.

Le modèle de APA est ensuite étendu et résolu de manière intégrée avec le problème d'attribution de zones de stockage, c'est-à-dire le problème d'attribution des différents types de chargement à des endroits spécifiques du port. Nous proposons un algorithme à solution exacte sophistiqué basé sur la méthode de branch-and-price pour résoudre le problème conjoint d'attribution de poste d'amarrage et de zones de stockage. Ce problème a été résolu en utilisant des méthodes métaheuristiques dans toutes les études précédentes traitant des terminaux à conteneurs. Les résultats basés sur des données réelles d'un port vraquier suggèrent que l'algorithme proposé peut être utilisé avec succès pour résoudre des instances de tailles réalistes dans un temps de calcul suffisamment raisonnable pour que l'algorithme puisse être implémenté et mis en pratique dans le port.

Un autre défi clé dans la planification des opérations portuaires est de prendre en compte la quantité énorme d'incertitude due à des facteurs comme les condi-

## Acknowledgments

---

tions météorologiques, les problèmes mécaniques et l'inefficacité de la main d'œuvre entre autres. Une perturbation stochastique peut rendre le respect des horaires irréalisable et, par conséquent, impliquer des coûts élevés pour le port. Dans la littérature actuelle, très peu d'études traitent de l'incertitude dans les opérations portuaires. Dans cette thèse, nous proposons des modèles et des algorithmes novateurs pour prendre en compte l'incertitude dans la planification. Ceux-ci sont basés sur des approches proactives et réactives.

Nous résolvons le problème d'attribution de postes d'amarrage sur horizon de planification mobile pour un horaire planifié de base et avec incertitude dans les heures d'arrivée et de manœuvre des navires. L'horaire est mis à jour en réponse à des perturbations, étant donné que les heures d'arrivée et de manœuvre des navires sont révélées en temps réel. Nous proposons des algorithmes d'ajustement basés sur la ré-optimisation et une approche gloutonne intelligente pour effectuer la réattribution et la replanification des navires, avec pour objectif de minimiser les coûts totaux engendrés par la planification modifiée de l'attribution des postes d'amarrage. L'incertitude liée à la partie de l'information qui doit encore être révélée est modélisée en faisant des hypothèses appropriées sur les distributions des paramètres incertains dérivés des données historiques. Les résultats suggèrent que la méthodologie que nous proposons peut réduire de manière significative les coûts engendrés, en comparaison à la pratique courante d'attribution des navires dans le port.

Afin de démontrer la complexité de la prise en compte de l'incertitude de manière proactive, nous effectuons une analyse théorique du problème de planification le plus élémentaire de la littérature, à savoir le problème de planification à une seule machine. Dans le contexte de la planification des opérations portuaires, le problème est analogue au problème discret d'attribution de poste d'amarrage, avec un seul postes d'amarrage. Dans toutes les études précédentes sur la planification robuste, l'incertitude liée aux temps de disponibilité de la tâche est largement ignorée. Nous intégrons l'incertitude dans les temps de disponibilité et de traitement des tâches, discutons les propriétés importantes de la planification robuste dans le contexte du problème de planification à une seule machine et proposons des heuristiques pour générer des horaires robustes.

Pour résumer, cette thèse propose des contributions fondamentales autant au niveau méthodologique qu'en termes des applications de la RO. En termes d'application, nous étudions les problèmes de décision survenant dans les terminaux de vrac et proposons des méthodes novatrices pour résoudre ces problèmes. Du point de vue méthodologique, nous traitons le problème de la prise en compte de l'incertitude dans les systèmes de planification logistique et de transport en particulier, et dans les problèmes de planification en général. Fina-



lement, la recherche présentée dans cette thèse ouvre la porte à plusieurs possibilités d'extension intéressantes et stimulantes, en particulier dans le domaine de la planification des opérations portuaires.

**Mots-clés** : logistique maritime, terminals de vracs, terminals à conteneurs, problèmes d'ordonnancement, programmation mixte en nombres entiers, métaheuristiques, génération de colonnes, optimisation sous incertitude.



# Contents

Acknowledgments	iii
Abstract (English/Français)	v
List of figures	xiii
List of tables	xvii
<b>1 Introduction</b>	<b>1</b>
1.1 Thesis Motivation and Objectives . . . . .	1
1.2 Thesis Contributions . . . . .	2
1.3 Thesis Outline . . . . .	4
 <b>I Bulk Terminal Management</b>	 <b>7</b>
<b>2 Bulk and Container Terminal Operations: A Comparative Analysis</b>	<b>9</b>
2.1 Introduction . . . . .	9
2.2 Decision Problems . . . . .	10
2.2.1 Stowage planning . . . . .	11
2.2.2 Berth Allocation Problem . . . . .	14
2.2.3 Crane Assignment and Scheduling Problem . . . . .	16
2.2.4 Transfer Operations . . . . .	21
2.2.5 Yard Management . . . . .	23
2.3 Conclusions . . . . .	27
 <b>3 The Deterministic Berth Allocation Problem in Bulk Ports</b>	 <b>29</b>
3.1 Introduction . . . . .	29
3.2 Literature Review . . . . .	32
3.3 Problem Definition . . . . .	34
3.4 Models for BAP . . . . .	37
3.4.1 MILP formulation . . . . .	37

## Contents

---

3.4.2	GSPP formulation . . . . .	40
3.4.3	SWO Heuristic . . . . .	43
3.5	Results and Analysis . . . . .	46
3.5.1	Generation of Instances . . . . .	46
3.5.2	Comparison of Algorithms . . . . .	52
3.5.3	Results Analysis . . . . .	56
3.6	Conclusions and Future Directions . . . . .	57
<b>4</b>	<b>The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports</b>	<b>63</b>
4.1	Introduction . . . . .	63
4.2	Literature Review . . . . .	65
4.3	Problem Statement . . . . .	67
4.4	Model Formulation . . . . .	71
4.4.1	Notation . . . . .	71
4.4.2	Mathematical Model . . . . .	72
4.5	Exact Solution Approach . . . . .	75
4.5.1	Set Partitioning Model . . . . .	76
4.5.2	Column Generation . . . . .	77
4.5.3	Branch and Bound . . . . .	83
4.6	Improvement Methods . . . . .	84
4.7	Meta-heuristic Approach . . . . .	86
4.8	Results and Discussion . . . . .	87
4.8.1	Generation of Instances . . . . .	87
4.8.2	Computational Results and Analysis . . . . .	89
4.9	Conclusions and Future Work . . . . .	95
<b>II</b>	<b>Handling Uncertainty</b>	<b>97</b>
<b>5</b>	<b>Real-Time Recovery in Berth Allocation</b>	<b>99</b>
5.1	Introduction . . . . .	99
5.2	Literature Review . . . . .	100
5.3	Problem Statement . . . . .	102
5.3.1	Baseline Schedule . . . . .	102
5.3.2	Real Time Recovery . . . . .	105
5.4	Recovery Methodology . . . . .	111
5.4.1	Modeling Uncertainty . . . . .	111
5.4.2	Solution Algorithms . . . . .	114
5.5	Results and Analysis . . . . .	121

5.5.1	Generation of Instances . . . . .	121
5.5.2	Comparison of Algorithms . . . . .	123
5.6	Conclusions and Future Work . . . . .	133
<b>6</b>	<b>The Robust Single Machine Scheduling Problem</b>	<b>135</b>
6.1	Introduction . . . . .	135
6.2	Literature Review . . . . .	137
6.3	Robust Single Machine Scheduling Problem . . . . .	139
6.3.1	Problem Definition . . . . .	139
6.3.2	Scheduling without release times . . . . .	141
6.3.3	Scheduling with release times . . . . .	142
6.4	Solution Algorithms to the ARSMSP with Release Times . . . . .	151
6.4.1	Iterated Local Search . . . . .	151
6.4.2	Variable Neighborhood Search Algorithm . . . . .	152
6.4.3	Methods to Determine the Benchmark Solution . . . . .	154
6.5	Computational Results and Analysis . . . . .	156
6.5.1	Generation of Instances . . . . .	156
6.5.2	Discussion of Results . . . . .	156
6.6	Conclusions and Future Work . . . . .	158
<b>7</b>	<b>Conclusions and Future Work</b>	<b>167</b>
7.1	Conclusions . . . . .	167
7.2	Future Research Perspectives . . . . .	169
	<b>Bibliography</b>	<b>184</b>



## List of Figures

2.1	<i>Development in international sea borne trade. Source: UNCTAD (2012)</i>	10
2.2	<i>Schematic diagram for port terminal operations (discharging operations</i>	11
2.3	<i>Stowage plan for bulk ports and container terminals . . . . .</i>	12
2.4	<i>Stowage planning and vehicle routing problem . . . . .</i>	13
2.5	<i>Quay side at a bulk port . . . . .</i>	15
2.6	<i>Container Quay Crane . . . . .</i>	17
2.7	<i>Mobile Harbor Crane used in Bulk Ports . . . . .</i>	17
2.8	<i>Bulk Pipeline . . . . .</i>	17
2.9	<i>Bulk Conveyor . . . . .</i>	17
2.10	<i>Straddle Carriers . . . . .</i>	22
2.11	<i>Automated Guided Vehicles . . . . .</i>	22
2.12	<i>Loading Shovel . . . . .</i>	22
2.13	<i>Wheel Loader . . . . .</i>	22
2.14	<i>Rubber Tyred Gantry Crane . . . . .</i>	24
2.15	<i>Rail Mounted Gantry Crane . . . . .</i>	24
3.1	<i>Berthing Layouts . . . . .</i>	31
3.2	<i>Feasible BAP Solution. . . . .</i>	31
3.3	<i>Schematic representation of vessels berthing along quay of length L for  N =3,  M =6 . . . . .</i>	35
3.4	<i>Schematic representation of a bulk port terminal . . . . .</i>	37
3.5	<i>Schematic Representation of SWO Algorithm . . . . .</i>	46
3.6	<i>Graphical Representation of the yard layout used in the generation of handling times . . . . .</i>	48
3.7	<i>Graphical Representation of the optimal solution output for the instance A1. The fixed facility locations are marked along the quay, where C and P stand for conveyor and pipeline respectively. . . . .</i>	57
3.8	<i>Graphical Representation of the optimal solution output for the instance C1. The fixed facility locations are marked along the quay, where C and P stand for conveyor and pipeline respectively. . . . .</i>	58

## List of Figures

---

3.9	<i>Effect of discretization and vessel lengths on service times for <math> N  = 10</math> vessels . . . . .</i>	60
3.10	<i>Effect of discretization and vessel lengths on service times for <math> N  = 25</math> vessels . . . . .</i>	60
3.11	<i>Effect of discretization and vessel lengths on service times for <math> N  = 40</math> vessels . . . . .</i>	60
3.12	<i>Effect of congestion on service times . . . . .</i>	61
4.1	<i>Schematic representation of a bulk port terminal . . . . .</i>	68
4.2	<i>Schematic representation of the port used in the test instances . . . . .</i>	90
5.1	<i>Hybrid berthing layout, showing a feasible assignment of vessels to berth sections at a single point in time . . . . .</i>	102
5.2	<i>Simple example of set partitioning to solve the BAP with <math> N  = 2</math>, <math> M  = 3</math> and <math> H  = 3</math> . . . . .</i>	103
5.3	<i>Baseline berthing positions for illustrative example with <math> N =3</math>, <math> M =6</math> and <math> H =4</math> . . . . .</i>	106
5.4	<i>Baseline schedule representing the mildly congested scenario . . . . .</i>	122
5.5	<i>Baseline schedule representing the highly congested scenario . . . . .</i>	122
5.6	<i>Comparison of the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between <math>t=H</math> and <math>t=2H</math> for the scenario with mild congestion and low stochasticity . . . .</i>	126
5.7	<i>Comparison of the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between <math>t=H</math> and <math>t=2H</math> for the scenario with mild congestion and high stochasticity . . .</i>	126
5.8	<i>Comparison of the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between <math>t=H</math> and <math>t=2H</math> for the scenario with high congestion and low stochasticity . . . .</i>	127
5.9	<i>Comparison of the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between <math>t=H</math> and <math>t=2H</math> for the scenario with high congestion and high stochasticity . . .</i>	127
5.10	<i>Comparison of the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between <math>t=H</math> and <math>t=2H</math> for the scenario with mild congestion and low stochasticity . . . .</i>	130
5.11	<i>Comparison of the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between <math>t=H</math> and <math>t=2H</math> for the scenario with mild congestion and high stochasticity . . .</i>	130
5.12	<i>Comparison of the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between <math>t=H</math> and <math>t=2H</math> for the scenario with high congestion and low stochasticity . . . .</i>	131



5.13	<i>Comparison of the compare the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between <math>t=H</math> and <math>t=2H</math> for the scenario with high congestion and high stochasticity . . . . .</i>	131
6.1	<i>Jobs <math>i</math>, <math>q</math> and <math>j</math> in an ordered sequence . . . . .</i>	147
6.2	<i>VNS Neighborhood Structures for a given sequence 1-2-3-4 . . . . .</i>	152
6.3	<i>VNS Neighborhood <math>N_1(1-2-3-4)</math> . . . . .</i>	152
6.4	<i>Binary string representations for the job sequence 1-2-3-4, for two different extreme point scenarios . . . . .</i>	155
6.5	<i>Convergence of Instance C18 over a computational time limit of 5 hours</i>	157
6.6	<i>Distribution of the output solution values for 50 simulation runs on instance C6 for a computational time limit of 300 seconds . . . . .</i>	158
6.7	<i>Distribution of the output solution values for 50 simulation runs on instance C12 for a computational time limit of 300 seconds . . . . .</i>	159
6.8	<i>Distribution of the output solution values for 50 simulation runs on instance C16 for a computational time limit of 300 seconds . . . . .</i>	159
6.9	<i>Distribution of the output solution values for 50 simulation runs on instance C19 for a computational time limit of 300 seconds . . . . .</i>	160



# List of Tables

2.1	A summary of the key features and research challenges in bulk terminals . . . . .	26
3.1	Assignment matrix for a simple example of GSPP . . . . .	41
3.2	Berthing layout and fixed facility locations for $ M =10$ (C and P stand for conveyor and pipeline respectively) . . . . .	49
3.3	Berthing layout and fixed facility locations for $ M =30$ (C and P stand for conveyor and pipeline respectively) . . . . .	50
3.4	Description of Instances . . . . .	51
3.5	Computational results for generated instances, Instances A, B, C .	54
3.6	Computational results for generated instances, Instances D, E, F .	55
4.5	Berthing layout and fixed facility positions for $ M =10$ (C and P stand for conveyor and pipeline respectively) . . . . .	89
4.6	Results obtained from the column generation for the instances containing $ N  = 10$ vessels. . . . .	91
4.7	Results obtained from the branch and bound for the instances containing $ N  = 10$ vessels. . . . .	92
4.8	Results obtained from the column generation for the instances containing $ N  = 25$ vessels. . . . .	93
4.9	Results obtained from the column generation for the instances containing $ N  = 40$ vessels. . . . .	93
4.10	Results obtained from CSNS for the instances containing $ N  = 10$ vessels. . . . .	94
4.11	Results obtained from CSNS for the instances containing $ N  = 25$ vessels. . . . .	94
4.12	Results obtained from CSNS for the instances containing $ N  = 40$ vessels. . . . .	95
5.1	Assignment matrix for a simple example of GSPP . . . . .	104
5.2	Baseline schedule for illustrative example with $ N =3$ , $ M =6$ and $ H =4$ . . . . .	106

## List of Tables

---

5.3	Disruption Scenario for a simple example with $ N =3$ , $ M =6$ and $ H =4$ . . . . .	107
5.4	A sample arrival disruption scenario for $ N =10$ vessels. ETA and ATA stand for the expected and actual arrival times respectively. In the arrival updates, the numbers indicate the updated arrival times and the numbers in the parantheses indicate the time instants at which the updates are received. . . . .	122
5.5	Performance indicators for the mildly congested scenario from $t=H$ to $t=2H$ averaged over 100 simulations runs . . . . .	128
5.6	Performance indicators for the highly congested scenario from $t=H$ to $t=2H$ averaged over 100 simulations runs . . . . .	128
5.7	Performance indicators for the mildly congested scenario from $t=H$ to $t=2H$ averaged over 100 simulations runs . . . . .	132
5.8	Performance indicators for the highly congested scenario from $t=H$ to $t=2H$ averaged over 100 simulations runs . . . . .	132
6.1	Assignment matrix for a simple example of set partitioning problem . . . . .	144
6.2	Computational results for generated instances with $ N =7$ . . . . .	161
6.3	Computational results for generated instances with $ N =15$ . . . . .	162
6.4	Computational results for generated instances with $ N =20$ . . . . .	163
6.5	Computational results for generated instances with $ N =30$ . . . . .	164
6.6	Computational results for generated instances with $ N =50$ . . . . .	165

# 1 Introduction

## 1.1 Thesis Motivation and Objectives

Transportation and logistics is one of the oldest application areas of operations research (OR). Classical combinatorial optimization problems in transportation, such as the vehicle routing problem, traveling salesman problem, and traffic flow problems, have contributed to fundamental knowledge in OR. Air transportation and the airline industry have also greatly benefited from the application of OR methods in a wide range of problems including airline scheduling, aircraft routing, fleet and crew assignment. The field of operations research has considerably evolved over the years, and the problems have become increasingly challenging and complex. Over the past few decades, the rapid growth of sea-freight transportation has pushed forward the need to apply the knowledge of operations research to the field of maritime transportation and seaport logistics. According to recent estimates, about 80% of the global trade by volume is carried by sea and handled by ports worldwide UNCTAD (2012). Maritime transportation perhaps represents the most essential form of transportation today, yet until recent years it has received limited attention in the OR literature within the academic community.

Over the years, the growth in the traded volume of containerized cargo has been much more dramatic as compared to bulk cargo, however the total traded volume of bulk cargo is still much higher than containerized cargo. In the existing operations research literature on port operations planning, the entire focus has been on addressing decision problems arising in container terminal operations. Bulk ports on the other hand have received almost no attention. The main objectives of the research presented in this thesis are two-fold. The first objective is to study how the existing work in the context of container terminals

can be extended to bulk ports, and in particular study the similarities and differences in applications and methodologies across these domains. We highlight the specific features of bulk port operations, and develop innovative models and solution algorithms to solve large scale optimization problems in bulk port operations planning.

The second objective is to address the challenge of handling uncertainty associated with port operations because of a variety of factors such as unfavorable weather conditions, mechanical problems, late arriving vessels and trucks etc. A small perturbation can possibly render a planned schedule infeasible, resulting in high losses. This necessitates the port authorities to adopt appropriate strategies to deal with uncertainty in information. In the past, very few studies related to port operations planning address this challenge. With an aim to fill this gap, we study the potential benefits of modeling uncertainty in port operations planning, and the added value of robustness in terms of system reliability and cost reduction. We study ways to handle uncertainty in real-time to minimize the impact of disruptions on the planned schedules, as well handle uncertainty in a proactive manner by incorporating uncertainty in the planning model with anticipation of variability in the available information.

## 1.2 Thesis Contributions

The main contributions of this thesis can be summarized as follows.

### *Part I: Bulk Terminal Management*

This thesis makes important fundamental contributions to the field of bulk port operations planning.

**Bulk port operations planning** We study the similarities and differences in applications and methodologies across the domains of container terminals and bulk ports, and devise solution algorithms specific to bulk ports. We study and solve the problem of allocating vessels along the quay, commonly referred to as the berth allocation problem in literature. We study the problem in isolation as well as in integration with the yard assignment problem, that concerns the assignment and storage of cargo types to specific locations on the yard. In the proposed optimization models, we explicitly account for the cargo type on the vessel and the fixed equipment facilities such as conveyors and pipelines which are installed at only certain sections along the quay. Our approach enhances

the co-ordination between the berthing and yard activities, apart from minimizing the total service cost of the berthing vessels. We add bulk-specific components to algorithms that have been previously used in the container terminal literature, including an exact method based on generalized set-partitioning, and metaheuristics based on squeaky wheel optimization and critical-shaking neighborhood search. The algorithms are tested and validated on instances inspired from real data from SAQR port, Ras Al Khaimah, UAE, the biggest bulk port in the middle east.

**Exact algorithms for large scale problems** In this thesis, we propose an exact solution algorithm based on a branch-and-price framework to solve the integrated problem of berth allocation and yard assignment in bulk ports. We highlight the specific features of bulk ports on both the seaside and the yard-side, and explicitly account for them in the proposed model. To the best of our knowledge, this is the first study to propose an exact solution approach to solve the combined problem, as all the previous studies in container terminals use metaheuristics to solve the problem, while in the context of bulk ports the problem has not been studied at all.

### *Part II: Handling Uncertainty*

In this thesis, we address the problem of handling uncertainty in scheduling problems, that can potentially render the planned schedules infeasible and result in high losses. We discuss both proactive and reactive strategies to model uncertainty to minimize the impact of disruptions, and maximize the system reliability and robustness.

**Reactive strategies to handle uncertainty** We study and solve the problem of recovering a baseline berthing schedule of vessels at a port in real time as disruptions occur. To the best of our knowledge, very few scholars have attempted to study the problem of real time recovery in port operations, which is typically based on local rescheduling heuristics or simple rules of thumb. We present an optimization based recovery algorithm based on set partitioning and a heuristic based smart greedy recovery method to solve the berth allocation problem on a rolling time horizon for a given baseline schedule, as the actual arrival and handling times of the vessels are revealed in real time. The solution performance of the algorithms is tested and validated by conducting a simulation studies in which the baseline schedule is the solution of the deterministic berth allocation problem. The results indicate that the modeling of uncertainty and

the proposed recovery algorithms can significantly lower the cost of reassigning the vessels in events of disruption as compared to the ongoing practice at the port.

**Proactive strategy to handle uncertainty** To permit useful analysis, we demonstrate the complexity in dealing with uncertainty in a proactive manner for the most basic scheduling problem - the single machine scheduling problem. In our problem, the release times and processing times of the jobs are specified as independent ranges of values with unknown probability distributions. The performance criterion is the total flow time of all the jobs and the robustness measure is the realized outcome for the worst-case contingency over the set of all potential scenarios. In previous research, the uncertainty in the release times of the jobs was largely ignored in the robust scheduling context. We illustrate the added complexity on considering non-zero release times, discuss properties of robust schedules and propose heuristic techniques based on variable neighborhood search and iterated local search to generate robust schedules.

### 1.3 Thesis Outline

In this section, we briefly outline the structure of this thesis. This thesis is organized in two main parts.

In *Part I* of this thesis, we present models and algorithms in the context of bulk port terminal management.

- **Chapter 2** presents a comparative analysis between the decision problems arising in container terminals and bulk ports from an OR perspective, provides a brief survey of the past literature on port operations planning and discusses several important open research problems in the bulk context.
- **Chapter 3** discusses two alternative exact solution methods based on mixed integer linear programming and generalized set partitioning, and a heuristic approach based on the principle of squeaky wheel optimization, to solve the deterministic berth allocation problem in bulk ports. This chapter has been published as:

Umang, N., Bierlaire, M. and Vacca, I. (2013). Exact and heuristic methods to solve the berth allocation problem in bulk ports. *Transportation Research*



*Part E: Logistics and Transportation Review*, 54: 14-31.

*Ranked 18th in the TOP 25 hottest articles of Transportation Research Part E: Logistics and Transportation Review for July-September 2013.*

- **Chapter 4** proposes an exact solution algorithm based on the branch-and-price framework to solve the integrated problem of berth allocation and yard assignment in bulk ports. This chapter has been published as:

Robenek, T., Umang, N., Bierlaire, M. and Ropke, S. (2014). A branch-and-price algorithm to solve the integrated berth allocation and yard assignment problem in bulk ports. *European Journal of Operational Research*, 235(2): 399-411.

*Part II* of the thesis focuses on developing reactive and proactive strategies to deal with uncertainty in scheduling problems.

- **Chapter 5** studies and solves the berth allocation problem on a rolling planning horizon with the objective to minimize the total realized costs of the updated berthing schedule as the actual arrival and handling time data is revealed in real time. This chapter is mainly based on:

Umang, N., Bierlaire, M. and Erera, A.L. (2013). Real-time management of the berth allocation problem with stochastic arrival and handling times, Technical report, TRANSP-OR, Ecole Polytechnique Fédérale de Lausanne.

- In **Chapter 6**, we illustrate the complexity in dealing with uncertainty in release times and processing times in a proactive manner for the single machine scheduling problem. This chapter is mainly based on:

Umang, N., Erera, A.L. and Bierlaire, M. (2013). The robust single machine scheduling problem with uncertain release and processing times, Technical report, TRANSP-OR, Ecole Polytechnique Fédérale de Lausanne.

- **Chapter 7** presents conclusions and discusses possible future directions of research.



# Bulk Terminal Management **Part I**



## 2 Bulk and Container Terminal Operations: A Comparative Analysis

In this chapter, we study the decision problems arising in port terminals from an operations research (OR) perspective. We carry out a comparative analysis in applications and methodologies across the domains of container terminals and bulk ports, and provide an overview of the existing literature. While on the one hand we focus on what extent the existing work on container terminals can be extended to bulk ports, on the other hand we identify the key issues specific to bulk terminals. The main objective is to provide a starting point for future research in bulk port operations planning, a research area that has been almost totally neglected in the OR literature thus far.

### 2.1 Introduction

The international sea borne trade registered an increase of 4% in the year 2011, with total volume of goods loaded worldwide reaching a record 8.7 billion tons (UNCTAD (2012)). While the container trade by volume increased by 8.6 %, the major bulks trade grew by 5.4 % and the world oil consumption increased marginally by 0.7 %. The plot in Figure 2.1 shows the development in international sea borne trade over the last four decades. It can be noted that while the growth in the container trade is the most rapid among all cargo types, the total traded volume of major bulks in 2012 is still much higher than containerized cargo.

The proper planning and management of port operations in view of the ever growing demand represents a big challenge. In general, the port terminal managers are faced with the challenge of maximizing efficiency both along the quay side and the yard. From the past research, it is well established that OR meth-

## Chapter 2. Bulk and Container Terminal Operations: A Comparative Analysis

ods and techniques can be successfully used to optimize port operations and enhance terminal efficiency. However while significant contributions have been made in the field of large scale optimization for container terminals, almost no attention has been directed to bulk ports. In the following sections, we discuss the decision problems arising in port terminals, present a brief survey of the past OR literature on port operations planning, and highlight the specific bulk port features that justify the need to devise models and solution algorithms specific to bulk ports.

### 2.2 Decision Problems

Port terminals typically have the following key operations that can be evaluated for port productivity.

1. stowage planning
2. berthing activities;
3. crane assignment and scheduling for loading or discharge;
4. quay to storage transfer;
5. yard storage;

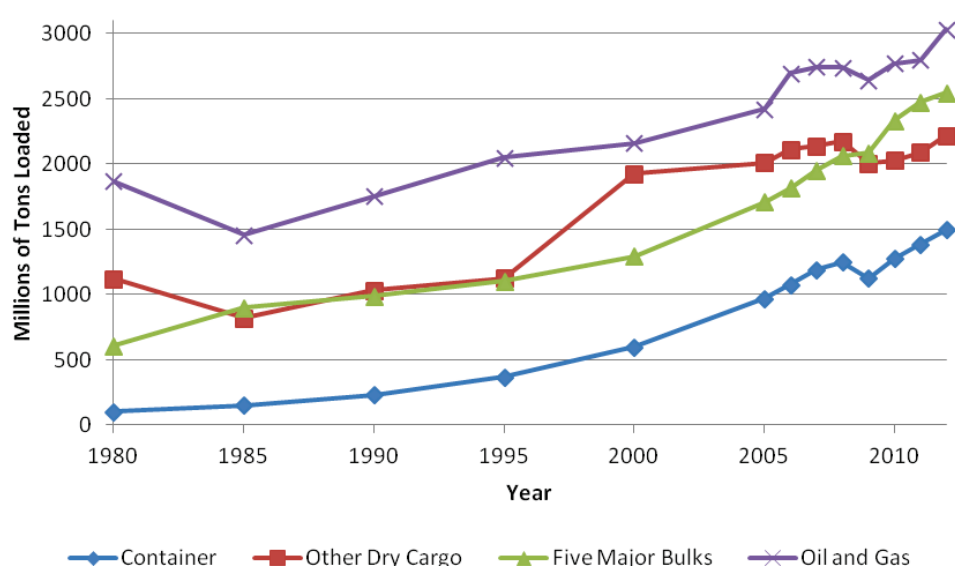


Figure 2.1: *Development in international sea borne trade. Source: UNCTAD (2012)*

### 6. intermodal transfer and inland distribution.

The operations are schematically shown in Figure 2.2 for vessel unloading. As evident, the cycle of operations is reversed in the case of loading operations. In practice, the cargo is transferred from the vessel to the quay side, and subsequently transferred from the quay to the yard. In bulk ports, depending on the cargo characteristics, the cargo may also be directly transferred from the vessel to the yard (or vice versa). For example, liquid bulk is generally discharged using pipelines that are installed at certain sections along the quay to oil tank terminals on the yard. Similarly rock aggregates may be directly loaded on the vessel using a conveyor facility from a factory outlet inside the port, without using any additional cranes.

In the following discussion, we focus on some of the key decision problems in port terminals. Most importantly we highlight some of the most important open research problems in the bulk context, that are worthy of investigation in future research.

For comprehensive literature surveys on container terminals, we refer the reader to Steenken et al. (2004), Stahlbock and Voss (2008), Bierwirth and Meisel (2010).

### 2.2.1 Stowage planning

The stowage planning problem involves the assignment of containers to specific holds within the vessel. In developing a stowage plan, many practical constraints related to the dimensions and the weight distribution of the containers need to be considered such that the vessel stability requirements and draft re-

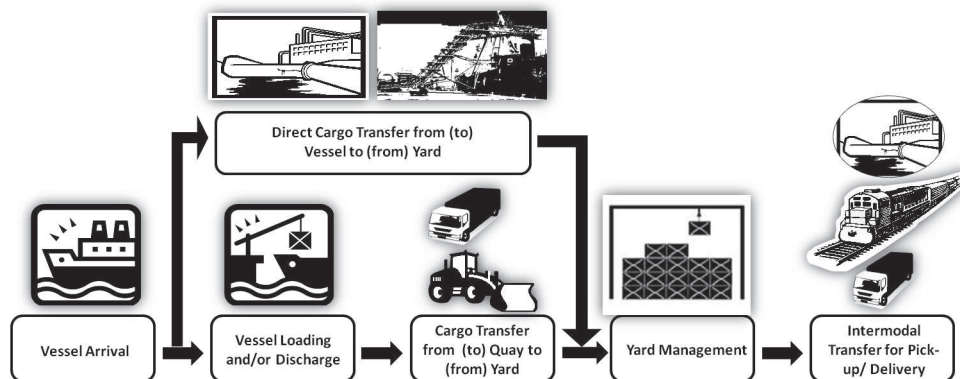


Figure 2.2: Schematic diagram for port terminal operations (discharging operations)

## Chapter 2. Bulk and Container Terminal Operations: A Comparative Analysis

strictions are met at each port of rotation of the vessel. In transshipment operations in container terminals, precedence relations among tasks can also be given to ensure that unloading precedes loading. There are several objectives such as the minimization of the total stowage or loading time, minimizing the number of shifts during port operations (vessel-to-vessel or vessel-to-quay), minimizing the number of container re-handles in yard stacks, maximizing vessel utilization and maximizing crane productivity. Ambrosino et al. (2006), Sciomachen and Tanfani (2007) and Imai et al. (2006) are few examples of such works.

In bulk ports, the weight distribution of the cargo needs to be considered in devising a stowage plan, while constraints related to the re-handling and stacking of cargo are mostly redundant. However restrictions on the adjacent storage of specific cargo brands are more stringent in the bulk context. For example, cargo brands such as coal and clay are typically not stored in the same or adjacent hatches of the vessel to prevent intermixing. This restriction makes it harder to devise feasible stowage plans that meet the vessel stability requirements at all ports of rotation of the vessel. For example, consider the case shown in Figure 2.3. The vessel has five holds to carry three cargo types A, B and C that have to be handled at three successive ports of rotation. In container terminals, this is usually not a problem since each hold of the vessel can store containers belonging to different cargo types, as shown. In the bulk context, to prevent intermixing, a single hold can usually carry at most a single cargo type, and the stowage plan shown may not be a feasible one, since after unloading the cargo type A at port 1, the vessel may become unstable.

An interesting research problem in this context deals with stowage planning and vessel routing to multiple ports of rotation. Consider a set of vessels that have to serve a given set of ports as shown in Figure 2.4. The objective is to allocate multiple brands of cargo to the individual holds of each vessel such that the total travel time or the total distance traveled by all vessels is minimized,

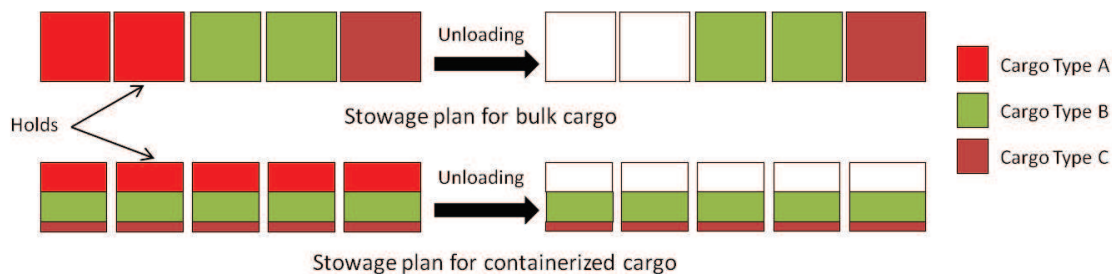


Figure 2.3: *Stowage plan for bulk ports and container terminals*



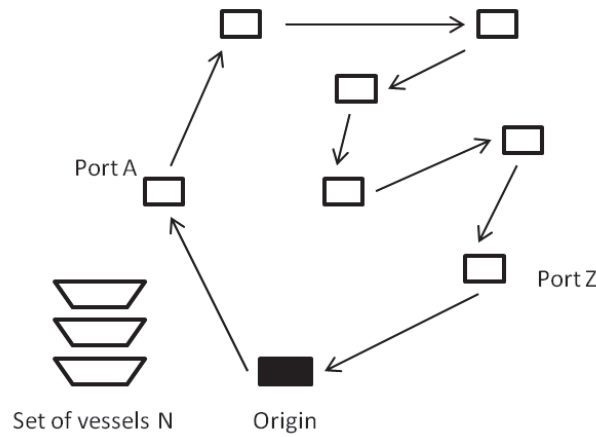


Figure 2.4: *Stowage planning and vehicle routing problem*

respecting the following set of constraints:

- the allocation should be done in a way such that the cargo demand in terms of the cargo brand and quantity is met at each port of rotation,
- the vessel should remain stable after the unloading operation at each port of rotation,
- the draft of the vessel, which is a function of the cargo weight on the vessel should be less than the draft of the port where the vessel is berthed,
- in the bulk context, each hold of the vessel may carry at most a single cargo type.

As evident, addressing the above problem entails combining ideas from stowage planning and vehicle routing problems in OR literature. Jenkins (2009) propose a mixed integer program (MIP) to solve a similar problem and reformulate the model using decomposition techniques including Benders decomposition and Dantzig Wolfe decomposition. However while the MIP formulation fails to generate good solutions for hard instances, the results for the proposed reformulation remain inconclusive. We believe this is an interesting open research problem that offers great scope to be addressed better in future research using more efficient exact methods such as branch-and-price and meta-heuristics for route generation.

### 2.2.2 Berth Allocation Problem

The berth allocation problem (BAP) in container terminals has been widely studied in the past. Imai et al. (1997), Imai et al. (2001), Imai et al. (2003), Cordeau et al. (2005), Monaco and Sammarra (2007), Mauri et al. (2008), Zhou and Kang (2008), Imai, Nishimura and Papadimitriou (2008), Han et al. (2010) and Buhrkal et al. (2011) propose methods to solve the discrete berth allocation problem. The continuous berth allocation problem is studied by Li et al. (1998), Lim (1998), Tong et al. (1999), Guan et al. (2002), Park and Kim (2002), Kim and Moon (2003), Park and Kim (2003), Guan and Cheung (2004), Imai et al. (2005) and Chang et al. (2008). The berth allocation problem with hybrid layout is addressed by Nishimura et al. (2001), Moorthy and Teo (2006), Dai et al. (2008) and Cheong et al. (2010), while position-dependent handling times are considered by Cordeau et al. (2005) and Imai et al. (2007) for indented berths.

We begin by presenting a mixed integer model to solve the continuous berth allocation problem in container terminals, originally proposed by Kim and Moon (2003).

#### Parameters

- $N$  : the total number of vessels
- $L$ : the length of a wharf
- $l_i$ : The length of vessel  $i$
- $p_i$ : The lowest cost berthing location of vessel  $i$
- $a_i$ : The estimated arrival time of vessel  $i$
- $d_i$ : The departure time requested by vessel  $i$
- $b_i$ : The time required for the ship operation for vessel  $i$
- $c_{1i}$ : The additional travel cost per unit distance for delivering the containers for vessel  $i$
- $c_{2i}$ : The penalty cost per unit time of vessel  $i$ , resulting from a departure delayed beyond the requested due time

#### Decision variables

- $x_i$ : The berthing position of vessel  $i$
- $y_i$ : The berthing time of vessel  $i$
- $z_{ij}^x$ : 1, if vessel  $i$  is located to the left of vessel  $j$  on the wharf; 0, otherwise
- $z_{ij}^y$ : 1, if vessel  $i$  is berthed before vessel  $j$ ; 0, otherwise
- $\alpha_i^+, \alpha_i^-, \beta_i^+, \beta_i^-$ : auxiliary decision variables

### Mathematical model

$$\min \sum_{i \in N} \{c_{1i}(\alpha_i^+ + \alpha_i^-) + c_{2i}\beta_i^+\} \quad (2.1)$$

$$\text{s.t. } x_i - p_i = \alpha_i^+ - \alpha_i^- \quad \forall i \in N \quad (2.2)$$

$$y_i + b_i - d_i = \beta_i^+ - \beta_i^- \quad \forall i \in N \quad (2.3)$$

$$x_i + l_i \leq L \quad \forall i \in N \quad (2.4)$$

$$x_i + l_i \leq x_j + M(1 - z_{ij}^x) \quad \forall i, j \in N, i \neq j \quad (2.5)$$

$$y_i + b_i \leq y_j + M(1 - z_{ij}^y) \quad \forall i, j \in N, i \neq j \quad (2.6)$$

$$z_{ij}^x + z_{ji}^x + z_{ij}^y + z_{ji}^y \geq 1 \quad \forall i, j \in N, i \neq j \quad (2.7)$$

$$y_i \geq a_i \quad \forall i \in N \quad (2.8)$$

$$\alpha_i^+, \alpha_i^-, \beta_i^+, \beta_i^-, x_i \geq 0 \quad \forall i \in N \quad (2.9)$$

$$z_{ij}^x, z_{ij}^y \in \{0, 1\} \quad \forall i, j \in N, i \neq j \quad (2.10)$$

In the above formulation,  $\alpha_i^+$  and  $\alpha_i^-$  are defined such that  $|x_i - p_i|$  is equal to  $\alpha_i^+$  when  $x_i - p_i \geq 0$ , and  $|x_i - p_i|$  is equal to  $\alpha_i^-$  when  $x_i - p_i \leq 0$ . Constraints (2.2) follow from this definition. Constraints (2.3) follow from similar definitions of  $\beta_i^+$  and  $\beta_i^-$ . Constraints (2.4) imply that the position of the rightmost end of

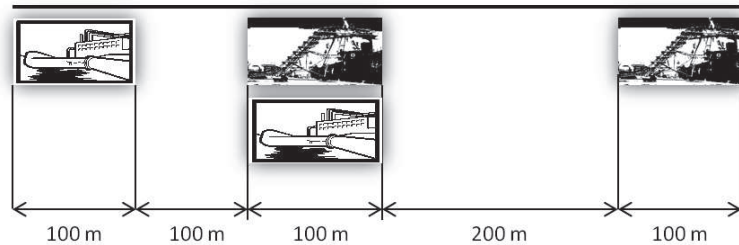


Figure 2.5: Quay side at a bulk port

## Chapter 2. Bulk and Container Terminal Operations: A Comparative Analysis

---

vessel  $i$  is restricted by the length of the wharf. Constraints (2.5)-(2.7) are the non-overlapping constraints that prevent two vessels  $i$  and  $j$  from occupying the same berthing location at the same time. Constraints (2.8) are the dynamic arrival constraints, that ensure that the vessel can berth only after it has arrived at the port.

While the constraints in the optimization model (2.1)-(2.10) hold in the bulk context, the model fails to capture some of the key features of bulk ports. In the bulk context, it is necessary to explicitly account for the cargo type on the vessel and the fixed facilities such as conveyors and pipelines that are installed at only certain sections along the quay, as shown graphically in Figure 2.5 for a quay length of 600 meters. This is because the choice of the transfer equipment including conveyors, pipelines and/ or mobile harbor cranes is determined by the specific cargo type on the vessel to be handled. For example liquid bulk may be discharged using pipelines only and rock aggregates may be loaded directly on the vessel using conveyors, without using any additional cranes. The dynamic, hybrid BAP in bulk ports has been recently studied by Umang et al. (2013), covered in Chapter 3 of this thesis. In this study, the handling times of the vessels are assumed to have a deterministic component depending on the number of operating cranes and the cargo type on the vessel, and a variable component that is a function of the relative berthing location of the vessel with respect to the storage location of the vessel cargo type on the yard. Since the conveyors and pipelines are used to handle specific cargo types, the specialized equipment facilities are also modeled as cargo types in the proposed model. A hybrid berthing layout is used to keep the model linear and ensure efficient space allocation of the vessels.

### 2.2.3 Crane Assignment and Scheduling Problem

In container terminals, the quayside operations are handled by quay cranes. In bulk ports, depending on the characteristics of the vessel and the cargo type, a wide variety of equipment is used for vessel loading and unloading operations. For example, there are specific cargo types for which only the conveyors and pipelines are used and the cranes may not be used at all. The loading/ discharging equipment used in port terminals is shown in Figures 2.6-2.9.

The problem of assigning and scheduling quay cranes for loading and discharging operations in container terminals is complex due to restrictions related to interference among cranes, and several other operational and precedence con-



Figure 2.6: Container Quay Crane



Figure 2.7: Mobile Harbor Crane used in Bulk Ports



Figure 2.8: Bulk Pipeline



Figure 2.9: Bulk Conveyor

straints. Lim et al. (2004), Kim and Park (2004), Moccia et al. (2006), Sammarra et al. (2007) and Bierwirth and Meisel (2009) are some of the works related to quay crane assignment and scheduling in container terminals. Daganzo (1989) discusses crane scheduling principles in port operations, and develops exact and approximate algorithms for crane scheduling in port terminals. We present a commonly used mixed integer formulation to solve the quay crane scheduling and assignment problem in container terminals, initially proposed by Sammarra et al. (2007) and later improved by Bierwirth and Meisel (2009) and more recently by Chen et al. (2014).

### Parameters

- $\Omega = \{1, 2, \dots, n\}$ : A set of tasks
- $Q = \{1, 2, \dots, q\}$ : A set of quay cranes
- $p_i$ : Processing time of task  $i$
- $l_i$ : Bay position of task  $i$

## Chapter 2. Bulk and Container Terminal Operations: A Comparative Analysis

---

- $\Phi$ : set of precedence constrained task pairs
- $\Psi$ : set of pairs of tasks that belong to adjacent bays
- $\hat{t}$ : The moving speed of a quay crane
- $t_{ij}$ : Traveling time between tasks  $i$  and  $j$
- $\delta$ : Safety margin between two adjacent quay cranes
- $r^k$ : Ready time for crane  $k$
- $l_0^k$ : Initial position for crane  $k$
- $0_k, T_k$ : two dummy tasks associated with crane  $k$ ,  $\{0_k \mid p_{0_k} = r^k, l_{0_k} = l_0^k\}_{k \in Q}$  and  $\{T_k \mid p_{T_k} = 0, l_{T_k} = l_T^k\}_{k \in Q}$
- $\overline{\Omega} : \Omega \cup \{0_k\}_{k \in Q} \cup \{T_k\}_{k \in Q}$ ,  $\Omega^{0_k} = \Omega \cup \{0_k\}$ ,  $\Omega^{T_k} = \Omega \cup \{T_k\}$ ,  $t_{ij} = \hat{t}|l_i - l_j|$  ( $i, j \in \overline{\Omega}$ )
- $\Delta_{ij}^{vw}$ : minimum time span to elapse between the processing of two tasks  $i$  and  $j$ , if processed by cranes  $v$  and  $w$ , respectively
- $\Theta : \{(i, j, v, w) \in \overline{\Omega}^2 \times Q^2 \mid (i < j) \wedge (\Delta_{ij}^{vw} > 0)\}$

### Decision variables

- $c_i$ :  $\geq 0$ , the completion of task  $i \in \overline{\Omega}$
- $c_{\max}$ :  $\geq c_i \forall i \in \{T_k\}_{k \in Q}$ , the makespan of the scheduling
- $x_{ij}^k$ : 1, if after handling task  $i$ , crane  $k$  is moving to handle task  $j$ ; 0, otherwise
- $z_{ij}$ : 1, if task  $i$  is handled before task  $j$ ; 0, otherwise

The objective function is the minimization of the makespan of the crane scheduling problem, represented by the decision variable  $c_{\max}$ . To ensure feasibility,

there are several additional operational and precedence constraints which are as follows.

$$\sum_{j \in \Omega^{T_k}} x_{0_k j}^k = 1 \quad \forall k \in Q \quad (2.11)$$

$$\sum_{j \in \Omega^{0_k}} x_{j T_k}^k = 1 \quad \forall k \in Q \quad (2.12)$$

$$\sum_{k \in Q} \sum_{j \in \Omega^{T_k}} x_{ij}^k = 1 \quad \forall i \in \Omega \quad (2.13)$$

$$\sum_{j \in \Omega^{0_k}} x_{ji}^k - \sum_{j \in \Omega^{T_k}} x_{ij}^k = 0 \quad \forall i \in \Omega, \forall k \in Q \quad (2.14)$$

$$c_i + t_{ij} + p_j - c_j \leq M(1 - x_{ij}^k) \quad \forall i, j \in \overline{\Omega}, \forall k \in Q \quad (2.15)$$

$$c_i + p_j - c_j \leq 0 \quad \forall i, j \in \Phi \quad (2.16)$$

$$c_i + p_j - c_j \leq M(1 - z_{ij}) \quad \forall i, j \in \overline{\Omega} \quad (2.17)$$

$$c_j - p_j - c_i \leq Mz_{ij} \quad \forall i, j \in \overline{\Omega} \quad (2.18)$$

$$z_{ij} + z_{ji} = 1 \quad \forall i, j \in \Psi \quad (2.19)$$

Constraints (2.11)-(2.12) ensure that each crane has exactly one task succeeding the first (dummy) task  $0_k$ , and exactly one task preceding the last (dummy) task  $T_k$ . Note that  $x_{ij}^k$  equal to 1, for  $i$  equal to  $0_k$  and  $j$  equal to  $T_k$ , implies that crane  $k$  remains idle. Constraints (2.13) state that every task is processed by exactly one quay crane. Constraints (2.14) state that for a given task  $i$  and crane  $k$ , there is exactly one task, including the dummy tasks, that is handled immediately before and after task  $i$ . Constraints (2.15) define the variables  $x_{ij}^k$ , constraints (2.16) are the precedence constraints, and constraints (2.17)-(2.18) define the variables  $z_{ij}$ . Constraints (2.19) ensure that tasks belonging to adjacent bays are not processed simultaneously.

The main point of difference between container terminals and bulk ports, is the set of non-interference constraints that forbid specific tasks from being processed simultaneously due to crane interference. In container terminals, the quay cranes move along guided rails and cannot pass each other. The set of all combinations of tasks and quay cranes that inevitably require a crossing of the assigned cranes and can thus possibly lead to crane interference is denoted by



## Chapter 2. Bulk and Container Terminal Operations: A Comparative Analysis

---

Θ. The non-interference constraints can be mathematically stated as follows.

$$\sum_{u \in \Omega^{0v}} x_{ui}^v + \sum_{u \in \Omega^{0w}} x_{uj}^w \leq 1 + z_{ij} + z_{ji} \quad \forall i, j, v, w \in \Theta \quad (2.20)$$

$$c_i + \Delta_{ij}^{vw} + p_j - c_j \leq M(3 - z_{ij} - \sum_{u \in \Omega^{0v}} x_{ui}^v - \sum_{u \in \Omega^{0w}} x_{uj}^w) \quad \forall i, j, v, w \in \Theta \quad (2.21)$$

$$c_j + \Delta_{ij}^{vw} + p_i - c_i \leq M(3 - z_{ji} - \sum_{u \in \Omega^{0v}} x_{ui}^v - \sum_{u \in \Omega^{0w}} x_{uj}^w) \quad \forall i, j, v, w \in \Theta \quad (2.22)$$

Constraints (2.20) prevent simultaneous processing of tasks  $i$  and  $j$  belonging to  $\Theta$ . For  $z_{ij} = 1$ , constraints (2.21) insert the minimum temporal distance between the completion time of task  $i$  and the starting time of task  $j$ . Similarly the case for  $z_{ji} = 1$  is handled by constraints (2.22).

The mobile harbor cranes used for loading and discharging operations in bulk ports can freely move around and pass each other, and thus the constraints (2.20)-(2.22) are redundant. In view of this, it is reasonable to assume a fixed crane deployment during the entire processing of a vessel, given that the port has enough cranes for them to be replaced in events of mechanical failure or breakdown. In the bulk context, the number of cranes operating on the vessel is usually a function of the length of the vessel and the cargo type on the vessel. Note that under the assumption of fixed crane deployment, the crane assignment problem is redundant, while the crane scheduling problem is expected to be easier to solve as compared to the quay crane scheduling problem in container terminals.

**Integrated Berth Allocation and Crane Assignment Problem** From the past OR literature on container terminal operations, it is well established that integrated planning of operations can allow port terminals to reduce congestion, lower delay costs and enhance efficiency. In container terminals, the integrated berth allocation and quay crane assignment and/or scheduling problem has been studied in the past by Park and Kim (2003), Meisel and Bierwirth (2006), Imai, Chen, Nishimura and Papadimitriou (2008), Meisel and Bierwirth (2009), Giallombardo et al. (2010) and Vacca (2011) among others. A comprehensive literature survey can be found in Bierwirth and Meisel (2010).

The integrated problem of berth allocation and crane scheduling in bulk ports differs from that in container terminals in the following respects:



- the need to explicitly account for the cargo type on the vessel,
- the presence of specialized facilities such as conveyors and pipelines at certain sections along the quay, and
- the redundancy of the non-interference restrictions on the cranes.

To address the integrated problem, the berth allocation model proposed by Umang et al. (2013) can be extended to explicitly account for the assignment of cranes to the vessels. Additional binary decision variables of the form  $x_{ijt}$  may be introduced in the model, which assume a value of 1 if crane  $j$  is assigned to vessel  $i$  at time  $t$ , and 0 otherwise. To ensure minimum separation requirements between the individual cranes operating on the same vessel, at any given time  $t$ , the number of cranes operating on the vessel cannot exceed  $L_i/l_s$  where  $L_i$  is the length of the vessel and  $l_s$  is the minimum safety distance to avoid interference. This can be mathematically expressed as  $\sum_{j \in J} x_{ijt} \leq L_i/l_s$ ,  $\forall i \in N, j \in J, t \in H$ , where  $N$ ,  $J$  and  $H$  are the sets of vessels, cranes and time steps in the planning horizon respectively.

In the model proposed by Umang et al. (2013), the handling times of the vessels have a deterministic component that is inversely proportional to the number of cranes operating on the vessel. The number of cranes operating on a given vessel is not fixed when the assumption of fixed crane deployment is relaxed. Thus in the integrated problem of berth allocation and crane scheduling, the modeling of the handling times may introduce non-linearities in the formulation, that may be linearized using standard techniques, and/or meta-heuristics need to be developed to solve realistic sized instances of the problem.

To the best of our knowledge, there is no existing work on the integrated problem of berth allocation and crane assignment and/or scheduling problem in the context of bulk ports, and we believe this to be another promising direction for future research.

### 2.2.4 Transfer Operations

In container terminals, the transfer operations are typically handled by straddle carriers, automated guided vehicles (AGVs) and internal trucks as shown in Figures 2.10-2.11. The objective in the optimization of transfer operations is usually to minimize the size of the vehicle fleet and/ or minimize the time taken for transfer. In container terminals, there have been a few studies related to

## Chapter 2. Bulk and Container Terminal Operations: A Comparative Analysis

---

the routing, scheduling, loading operations and vehicle deployment of internal trucks, straddle carriers and AGVS's. Liu et al. (2004), Vis et al. (2005) and Cheng et al. (2005) are few examples of related works.

In bulk ports on the other hand, dry bulk cargo is typically transferred from the quay side to the yard (or vice versa) using a wide variety of equipment such as loading shovels and wheel loaders as shown in Figures 2.12-2.13, mini loaders and internal trucks. As discussed earlier, there are also specialized facilities such as conveyors and pipelines that are installed at certain sections along the quay and that may be used to directly transfer the cargo from the vessel to the yard (or vice versa). To the best of our knowledge, almost no work has been done on the routing and scheduling operations of the transfer equipment in the bulk context, which can significantly enhance the terminal throughput. There are also several other open research questions specific to bulk ports, related to the design and operation of conveyors and pipelines, that need to be addressed in future studies.



Figure 2.10: *Straddle Carriers*



Figure 2.11: *Automated Guided Vehicles*



Figure 2.12: *Loading Shovel*



Figure 2.13: *Wheel Loader*

### 2.2.5 Yard Management

In container terminals stacking operations in the yard blocks are carried out by rubber tyred or rail mounted gantry (RTG/ RMG) cranes and straddle carriers as shown in Figures 2.14-2.15. Yard management in container terminals involves several tactical and operational level decision problems. Scheduling of yard cranes is addressed by Cheung et al. (2002), Zhang et al. (2002), Ng and Mak (2005), Ng (2005) and Jung and Kim (2006). Storage and space allocation, stacking and re-marshaling strategies have been studied by Kim and Kim (1999), Kim et al. (2003), Lee et al. (2006) and few others. Nishimura et al. (2009) investigate the storage plan for transshipment hubs, and propose an optimization model to minimize the sum of the waiting time of feeders and the handling times for transshipment containers flow. The cost-effective management of empty containers considering deterministic systems has been studied by Erera et al. (2005), Shintani et al. (2007) and many others, while the stochastic nature of the problem has been addressed by Crainic et al. (1993), Erera et al. (2009) among others.

In bulk ports, the management of yard operations involves a wide range of decision problems in accordance with the cargo characteristics, such as routing and scheduling of internal trucks and auxiliary equipment for transfer of cargo within the yard, and storage allocation of multiple brands of cargo on the yard. Dry bulk cargo is stored in a variety of enclosures or open yard configurations. Liquid bulk cargo is stored in tank terminals. The storage component for both dry and liquid bulk cargo can also include other value-added activities such as blending or processing. There are strict restrictions forbidding specific brands of cargo from being stored at adjacent locations in the yard. While the existing literature on yard management in the context of bulk ports is extremely scarce, there have been a few recent studies on the optimization of the Hunter Valley Coal Supply Chain and the effective yard management at the Port of Newcastle (Boland et al. (2012), Singh et al. (2012) and few others), the world's largest coal export port. These papers develop stockyard planning technology and study cost-effective capacity improvement initiatives using techniques such as integer programming and greedy heuristics to meet the anticipated rise in the coal demand in the future.

**Integrated Berth Allocation and Yard Assignment Problem** In the context of container terminals there have been few studies related to the integrated planning of berth and yard activities. Moorthy and Teo (2006) discuss the concepts of berth template and yard template in the context of transshipment hubs in

## Chapter 2. Bulk and Container Terminal Operations: A Comparative Analysis

---



Figure 2.14: *Rubber Tyred Gantry Crane*



Figure 2.15: *Rail Mounted Gantry Crane*

container shipping and develop a robust berth allocation plan using the sequence pair approach. Cordeau et al. (2007) develop an evolutionary heuristic to study the Service Allocation Problem, a tactical problem arising in the yard management of Gioia Tauro Terminal. Zhen, Chew and Lee (2011) and more recently Lee and Jin (2013) are other examples of works that study and simultaneously solve the tactical berth template planning in combination with quay crane assignment and yard template planning in container transshipment hubs. Other works on integrated problems related to yard management in container terminals include Bish et al. (2001) and Kozan and Preston (2006) who propose the integration of yard allocation and container transfers, and Chen et al. (2007) and Lau and Zhao (2007) who study the integrated scheduling of handling equipment in a container terminal.

To the best of our knowledge, the only existing study in the field of integrated planning of operations in the context of bulk ports is by Robenek et al. (2014), covered in Chapter 4 of this thesis. The authors study and solve the combined large scale problem of berth allocation and yard assignment assuming a fixed crane deployment. The proposed model seeks to determine the optimal assignment of vessels to specific berthing locations, and that of cargo brands to specific yard locations, for which the total service times of the berthing vessels is minimized. The model captures specific bulk features including the presence of fixed equipment facilities such as conveyors and pipelines, and restrictions on the storage of specific cargo brands, as discussed earlier. An exact solution algorithm based on the branch-and-price framework and a meta-heuristic based on critical shaking neighborhood search are proposed to solve the integrated problem.

In yard management in bulk ports, there are several challenging problems that have yet to be addressed that can make a significant contribution to the existing literature on port operations planning. While the model proposed by Robenek et al. (2014) integrates berth allocation with yard assignment, it does not explicitly model the deployment, routing and scheduling of the wide variety of equipment used for transfer operations. Some of the open research problems include the integration of the deployment of transfer equipment and berth allocation for a given yard layout, and a three-level planning problem including berth allocation, deployment of transfer equipment and the yard assignment problem. Due to the large scale nature of these problems, the corresponding mixed integer formulations are extremely complex and unwieldy. Therefore large scale optimization techniques based on meta-heuristics and/or state-of-the-art exact methods such as branch-and-price or branch-and-cut should be developed to address these problems.

Based on the preceding discussion, in the following table we briefly summarize the key bulk features and the associated research challenges in bulk ports. We also cite references of existing works focusing on bulk ports.



Decision Problem	Specific Bulk Features	Research Challenges and Key References
Stowage Planning	<ol style="list-style-type: none"> <li>1. Constraints related to stacking and re-handling of cargo are redundant.</li> <li>2. Restrictions on storage location of specific cargo types within the vessel are more stringent.</li> </ol>	<ol style="list-style-type: none"> <li>1. Devising a stowage plan that satisfies the vessel stability requirements and draft restrictions at multiple ports of rotation.</li> </ol>
Berth Allocation Problem	<ol style="list-style-type: none"> <li>1. The cargo type on the vessel needs to be explicitly considered.</li> <li>2. Fixed equipment facilities such as conveyors and pipelines need to be explicitly modeled.</li> <li>3. A hybrid berthing layout is most efficient.</li> </ol>	<ol style="list-style-type: none"> <li>1. Umang et al. (2013) study the dynamic, hybrid BAP in bulk ports.</li> </ol>
Crane Scheduling Problem	<ol style="list-style-type: none"> <li>1. Mobile harbor cranes used in bulk ports can freely move around and pass each other.</li> <li>2. The assumption of fixed crane deployment is more reasonable.</li> </ol>	<ol style="list-style-type: none"> <li>1. The integrated problem of berth allocation and crane scheduling in the bulk context.</li> </ol>
Transfer Operations	<ol style="list-style-type: none"> <li>1. A wide variety of equipment such as internal trucks, loading shovels, mini loaders and wheel loaders are used.</li> <li>2. The cargo may be directly transferred from the vessel vessel to the yard (or vice versa) using fixed facilities such as conveyors and pipelines.</li> </ol>	<ol style="list-style-type: none"> <li>1. Routing and scheduling of transfer equipment.</li> <li>2. Design and operation of conveyors and pipelines.</li> </ol>
Yard Management	<ol style="list-style-type: none"> <li>1. Restrictions on the storage of specific cargo types is more stringent.</li> <li>2. Yard storage management of the cargo is typically done by transfer equipment including trucks and auxiliary equipment facilities.</li> </ol>	<ol style="list-style-type: none"> <li>1. Robenek et al. (2014) study the integrated berth allocation and yard assignment problem in bulk ports.</li> <li>2. Optimization of the Hunter Valley Coal Supply Chain by Boland et al. (2012), Singh et al. (2012) and few others.</li> <li>3. The combined problem of berth allocation and deployment of transfer equipment with or without integration with yard assignment represents a promising direction for future research.</li> </ol>

Table 2.1: A summary of the key features and research challenges in bulk terminals

## **2.3 Conclusions**

We have carried out a comparative analysis of the decision problems arising in container terminals and bulk ports from an OR perspective. We conclude that while there are many similarities in applications and methodologies across these domains, there are also several differences that call for the need to devise models and methods specific to bulk ports operations planning. In this chapter, we have provided a brief literature survey of the decision problems in port operations planning, with a special focus on the future research challenges in the bulk context.





## 3 The Deterministic Berth Allocation Problem in Bulk Ports

In this chapter, we study the dynamic hybrid berth allocation problem in bulk ports with the objective to minimize the total service times of the vessels. We propose two exact methods based on mixed integer programming and generalized set partitioning, and a heuristic method based on squeaky wheel optimization, explicitly considering the cargo type on the vessel. The formulations are compared through extensive numerical experiments based on instances inspired from real bulk port data. The results indicate that the set partitioning method and the heuristic method can be used to obtain near-optimal solutions for even larger problem size.

### 3.1 Introduction

The Berth Allocation Problem (BAP) refers to the problem of assigning a set of vessels to a given berth layout within a given time horizon. There could be several objectives such as minimization of the service times to vessels, minimization of port stay time, minimization of number of rejected vessels, minimization of deviation between actual and planned berthing schedules etc. There are several spatial and temporal constraints involved in the BAP, which lead to a multitude of BAP formulations. The existing models for BAP in literature can be classified on the basis of the temporal attributes such as vessel arrival process, start of service and handling times of vessels as well as spatial attributes relating to the berth layout, draft restrictions and others.

According to Bierwirth and Meisel (2010), the vessel arrival process can be considered as static or dynamic. In the static case, the arrival times do not impose a hard constraint on the berthing times, and vessels can berth at any time given

that the allocated portion of the quay is available for berthing. In the dynamic variant of the problem, vessels cannot berth before they have arrived at the port. The vessel arrivals can be deterministic in which fixed expected values of arrival times are given, or stochastic in which a distribution of arrival times may be given to account for uncertainty in vessel arrival times.

The handling times for vessels can be assumed as fixed and unchangeable, or dependent on the berthing positions of vessels and/ or work schedule and number of cranes assigned to vessels. The handling times may also be considered as stochastic to account for uncertainty in handling times due to unforeseen disruptions such as equipment breakdown or unavailability of equipment or cargo due to any other reason.

Spatial constraints limit the feasible berthing positions of vessels according to a preset partitioning of the quay into berths. On the basis of berth layout, the BAP can be classified as discrete, continuous or hybrid (Bierwirth and Meisel (2010)). In the discrete case, the quay is divided into a set of sections or berths, and a given berth can be used by at most one vessel at any given time. In the continuous case, there is no partitioning of the quay, and a vessel can occupy any arbitrary position along the quay. This understandably leads to better utilization of the quay space, but is computationally more complicated. In the hybrid case, the quay is partitioned into a set of sections, but a vessel can occupy more than one section at a time, and more than one vessel may be allowed to share the same section at the same time. A graphical representation of different berth layouts is shown in Figure 3.1. In addition, the draft restrictions on vessels which limit the feasible berthing positions of vessels to only those berths or sections which have a draft higher than the draft of the vessel may also be considered in formulating the BAP.

A feasible berthing assignment can be represented on a space-time graph as shown in Figure 3.2. The vertical axis corresponds to the quay space within the quay boundary, while the horizontal axis represents berthing time within the planning horizon. Each rectangle represents a vessel berthing at the port. The height of the rectangle represents the length of the vessel, with the upper and lower co-ordinates indicating the berthing location along the quay. The width of the rectangle represents the handling or processing time of the vessel, with the left and right co-ordinates indicating the start and end of handling time respectively. While two vessels may be overlapping in space or in time, it is infeasible for two vessels to overlap in both space and time simultaneously. Thus, in a feasible berthing assignment, all rectangles (vessels) should be non-overlapping

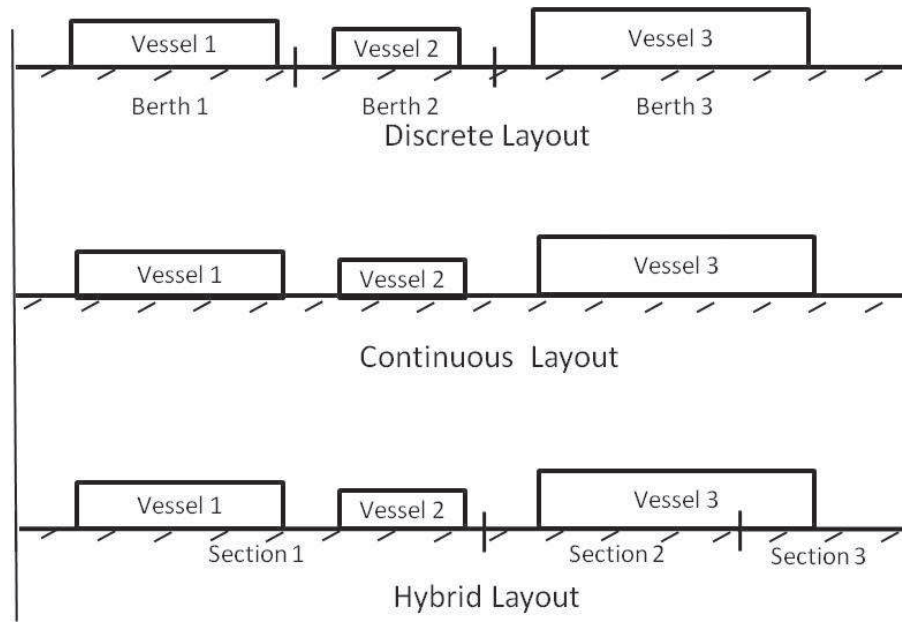


Figure 3.1: *Berthing Layouts*

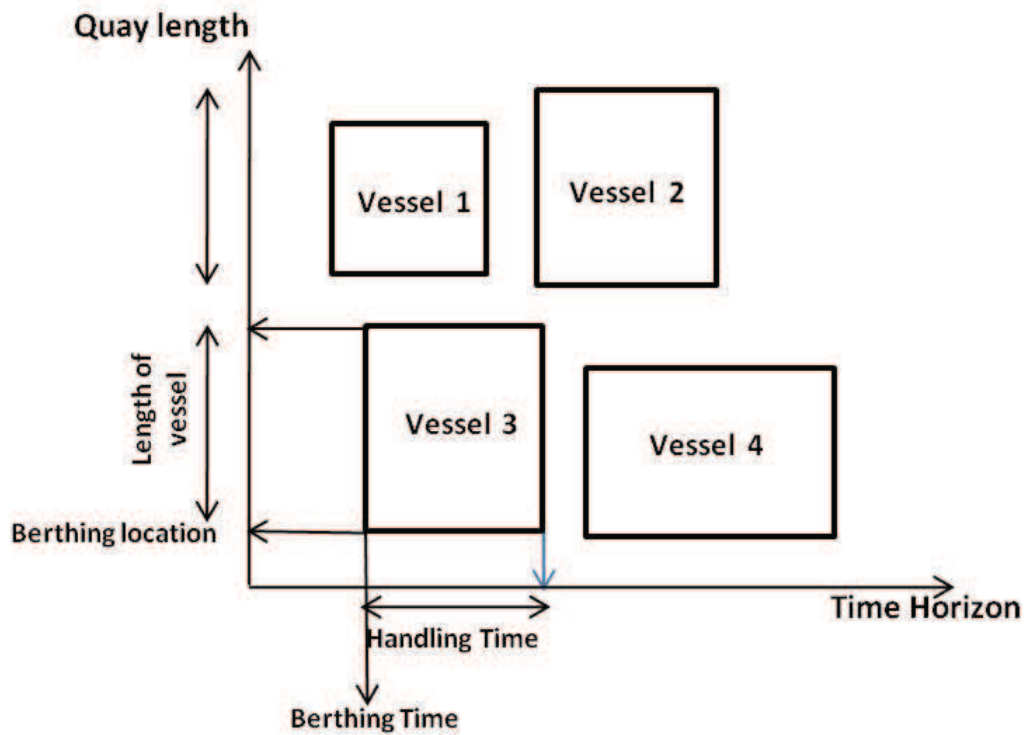


Figure 3.2: *Feasible BAP Solution.*

and each individual vessel should respect the spatial and temporal constraints

on its berthing. This representation of the BAP on the time-space graph further makes it possible for scholars to study the berth allocation problem as a 2-D bin packing problem (Lim (1998)).

In this research, we discuss the dynamic, hybrid berth allocation problem in bulk ports. To the best of our knowledge, this is the first study on the berth allocation problem in the context of bulk ports, discussing BAP formulations that explicitly take into account the cargo type on the vessel. A mixed integer linear programming approach is presented to solve the problem. An alternative exact solution algorithm based on generalized set partitioning approach is presented to solve larger and more complex instances of the BAP under study. To solve the problem in large scale realistic environments, a heuristic algorithm based on an optimization approach, commonly termed in literature as squeaky wheel optimization is also developed (Clements et al. (1997)). This technique has been successfully applied in graph coloring and scheduling problems, and more recently used to solve operations research problems in container terminals by few scholars such as Fu et al. (2007) and Meisel and Bierwirth (2009). The approach is adapted to solve the hybrid, dynamic BAP in bulk ports. Numerical experiments are conducted on instances based on real port data to test and validate the efficiency of the proposed algorithms.

### 3.2 Literature Review

In this section we present a brief review of past literature on the berth allocation problem in the context of container terminals.

**Discrete BAP.** The static variant of discrete BAP has been studied by Imai et al. (1997) which minimizes the total service times of vessels and the deviation between arrival order and service order of vessels, Imai et al. (2001) and Imai, Nishimura and Papadimitriou (2008)). The dynamic discrete BAP problem is considered by Imai et al. (2001), Imai et al. (2003), Monaco and Sammarra (2007), Buhrkal et al. (2011) and few others. More recent approaches, such as Zhou and Kang (2008) and Han et al. (2010), solve the problem considering stochasticity in both arrival times and handling times of vessels. Cordeau et al. (2005) uses a tabu search method to solve the discrete dynamic BAP with due dates, which is further improved upon by Mauri et al. (2008) using a column generation approach that delivers higher quality solutions in lesser computation time. Vacca (2011) study the discrete dynamic BAP at the tactical level in integration with the quay crane scheduling problem, and propose a two-level heuristic to solve

the problem.

**Continuous BAP.** The static continuous BAP has been considered by Li et al. (1998), Guan et al. (2002) and Park and Kim (2003). Guan and Cheung (2004) consider continuous dynamic BAP with fixed handling times using a tree search procedure to minimize the total weighted port stay time of vessels. Gao et al. (2010) use a robust planning approach to solve a dynamic continuous BAP with stochastic vessel arrivals via feedback procedure in the planning stage. Minimization of tardiness as an objective in continuous dynamic BAP is considered by Park and Kim (2002) using a sub-gradient method and by Kim and Moon (2003) using simulated annealing approach. Minimization of quay length with given berthing times as an objective is studied by Lim (1998) and Tong et al. (1999). The continuous BAP with handling times depending on berthing positions is studied by Imai et al. (2005) and Chang et al. (2008) who further considers draft restrictions in the BAP model.

**Hybrid BAP.** The dynamic hybrid BAP with fixed handling times is considered by Moorthy and Teo (2006), which considers a robust planning approach by incorporating stochasticity in vessel arrivals, and further studied by Dai et al. (2008). The dynamic hybrid BAP with position-dependent handling times is studied by Imai et al. (2007) for indented berths, and Cordeau et al. (2005). Draft restrictions in dynamic hybrid BAP are considered by Nishimura et al. (2001) and Cheong et al. (2010).

Comprehensive literature surveys on the BAP in context of container terminal operations can be found in Bierwirth and Meisel (2010), Steenken et al. (2004) and Stahlbock and Voss (2008). To our knowledge, the problem has not been investigated thus far in the context of bulk port terminals, which is the primary focus of this research. Unlike container terminals, in bulk ports it is necessary to account for the cargo type on the vessel and model the interaction between the yard layout concerning the location of specific cargo types on the yard and the berthing locations of the vessels. Moreover, in container terminals the loading and unloading operations are usually carried out using rail mounted gantry (RMG) cranes that move along a guided rail and cannot pass each other. On the other hand in bulk terminals there is a wide range of heterogeneous loading/unloading equipment. This includes specialized equipment facilities such as conveyors and pipelines that are installed at only certain sections along the quay, and mobile harbor cranes that can be freely moved around and can pass each other during the service of a vessel. These differences among others necessitate the need to devise specific solutions for bulk ports. In this study, we

discuss exact and heuristic algorithms to solve the berth allocation problem in the context of bulk ports and compare the algorithms from a computational perspective based on instances inspired from real bulk port data.

### 3.3 Problem Definition

We consider a set of vessels  $N$ , to be berthed on a continuous quay of length  $L$  for a time horizon  $H$ . We consider dynamic vessel arrival process and a berth layout which is an extension of the hybrid case. We discretize the quay boundary into a set  $M$  of sections of variable lengths. In a feasible berthing assignment, a given vessel may occupy more than one section, however a given section cannot be occupied by more than one vessel or part of a vessel at any given time. Partitioning the quay space into sections of variable length brings more flexibility to the model, and the manner in which sections are defined along the quay is critical.

One major difference that distinguishes the Berth Allocation Problem (BAP) in bulk ports from that in container terminals is the presence of fixed specialized equipment facilities such as conveyors and pipelines at bulk ports. In a container terminal, all cargo is packed into containers, and thus there is no need for any specialized equipment to handle any particular type of cargo. In contrast in bulk ports, depending on the vessel requirements and cargo properties, a wide variety of equipment is used for discharging or loading operations. For example, liquid bulk is generally discharged using pipelines which are installed at only certain sections along the quay. Similarly, a vessel may require the conveyor facility to load cargo from a nearby factory outlet to the vessel. For a given vessel, the handling time has a variable component as determined by the berthing position of the vessel along the quay and a fixed component determined by the number of quay cranes operating on the vessel. The berthing position of the vessel along the quay, determines the distance between the berthing position and the storage location of the cargo type of the vessel on the yard. This in turn determines the time taken to transfer cargo between the berthing location and the cargo location on the yard using auxiliary equipment facilities such as loading shovels, trucks etc. or specialized facilities such as conveyors and pipelines.

We define a single variable co-ordinate system along the quay, with the origin at the left extreme of the quay. The vessels berth from the beginning of the first occupied section. This is schematically shown in Figure 3.3 for  $|N| = 3$  and  $|M|$

= 6.

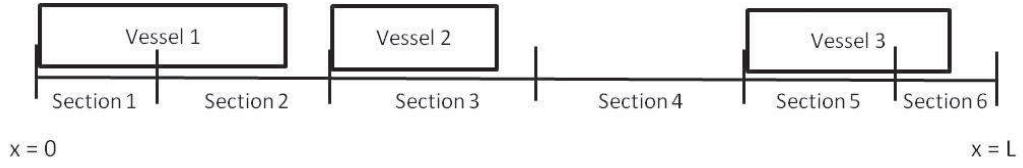


Figure 3.3: Schematic representation of vessels berthing along quay of length  $L$  for  $|N|=3$ ,  $|M|=6$

To model the hybrid, dynamic berth allocation problem (BAP) in bulk ports, we assume the following input data to be available:

- $N =$  set of vessels
- $M =$  set of sections
- $k = 1, \dots, |M|$  sections along the quay
- $i = 1, \dots, |N|$  vessels berthing at the port
- $A_i =$  expected arrival time of vessel  $i$
- $D_i =$  draft of vessel  $i$
- $L_i =$  length of vessel  $i$
- $Q_i =$  quantity of cargo for vessel  $i$
- $W_i =$  cargo type to be loaded or discharged from vessel  $i$
- $d_k =$  draft of section  $k$
- $\ell_k =$  length of section  $k$
- $b_k =$  starting coordinate of section  $k$
- $h_k^w =$  handling time for unit quantity of cargo type  $w$  for vessel berthed at section  $k$
- $L =$  total length of quay

The clearance distances between adjacent vessels as well as end-clearances may be considered implicitly in vessel lengths. Similarly, the clearance times between two successive vessels overlapping in space may be considered implicitly in the handling times

In the computation of handling times, the main assumption is that all sections occupied by the berthed vessel are being operated simultaneously with each section handling the amount of cargo proportionally to the section length. The handling time of the vessel is the time taken to load or discharge the section whose operation finishes last. The unit handling time  $h_k^w$  for section  $k$  and cargo type  $w$  includes the time taken to transfer unit quantity of cargo between the cargo location on the yard and the berthed section, and the time taken to load



(or unload) the cargo from the quay side to the vessel. In equation 3.1, these have been denoted by  $\beta_k^w$  and  $\alpha_k^w$  respectively. Thus we have,

$$h_k^w = \alpha_k^w + \beta_k^w \quad (3.1)$$

$$\alpha_k^w = T/n_k^w \quad (3.2)$$

$$\beta_k^w = v^w d_k^w \quad (3.3)$$

In equation 3.2,  $T$  is the amount of time taken by a single crane to load or discharge a unit quantity of cargo, and  $n_k^w$  is the number of cranes operating in section  $k$  for cargo type  $w$ .  $\beta_k^w$  is the time taken to transfer a unit quantity of cargo between the location of cargo type  $w$  on the yard and the section  $k$ , which is assumed to be a linear function of the distance  $d_k^w$  between the two locations. The parameter  $v^w$  depends on the rate of transfer of cargo type  $w$ . A schematic representation of a bulk port terminal is shown in Figure 3.4. As shown, for a vessel carrying cement ( $w=8$ ) berthed at section  $k=5$ , the unit handling time value is  $h_5^8 = \alpha_5^8 + \beta_5^8 = T/n_5^8 + v^8 d_5^8$ , where  $d_5^8$  is the distance between the section  $k=5$  and the cargo location  $w=8$ ,  $v^8$  is a function of the rate of transfer of cement from the cargo location to the berthed section and  $n_5^8$  is the number of cranes operating in the section. Alternatively, if a vessel is using the conveyor facility to load rock aggregates from the rock factory directly into the vessel, the vessel must occupy section  $k=4$ . If no additional cranes are being used to transfer the cargo at  $k=4$ , we provide  $\alpha_4^{11} \rightarrow 0$ , and  $h_4^{11} = \beta_4^{11} = v^{11} d_4^{11}$ , where  $v^{11}$  is a parameter dependent on the rate of material transfer in the conveyor and  $d_4^{11}$  is the distance between the rock factory and the conveyor facility.

In our model, the specialized equipment facilities such as conveyors and pipelines are also modeled as cargo types. The modeling of handling times using specialized facilities is taken into account by generating appropriate values of unit handling times  $h_k^w$  for every section  $k$  that may or may not have the cargo facility  $w$ . For example, if a particular cargo type  $w$  corresponding to the conveyor facility cannot be handled by section  $k$  which does not have this facility, the unit handling time value  $h_k^w$  is set to infinity or an extremely large value, for this particular section and cargo facility. This prevents the vessels needing the conveyor facility from berthing at any section(s) other than the section(s) where the facility is installed.



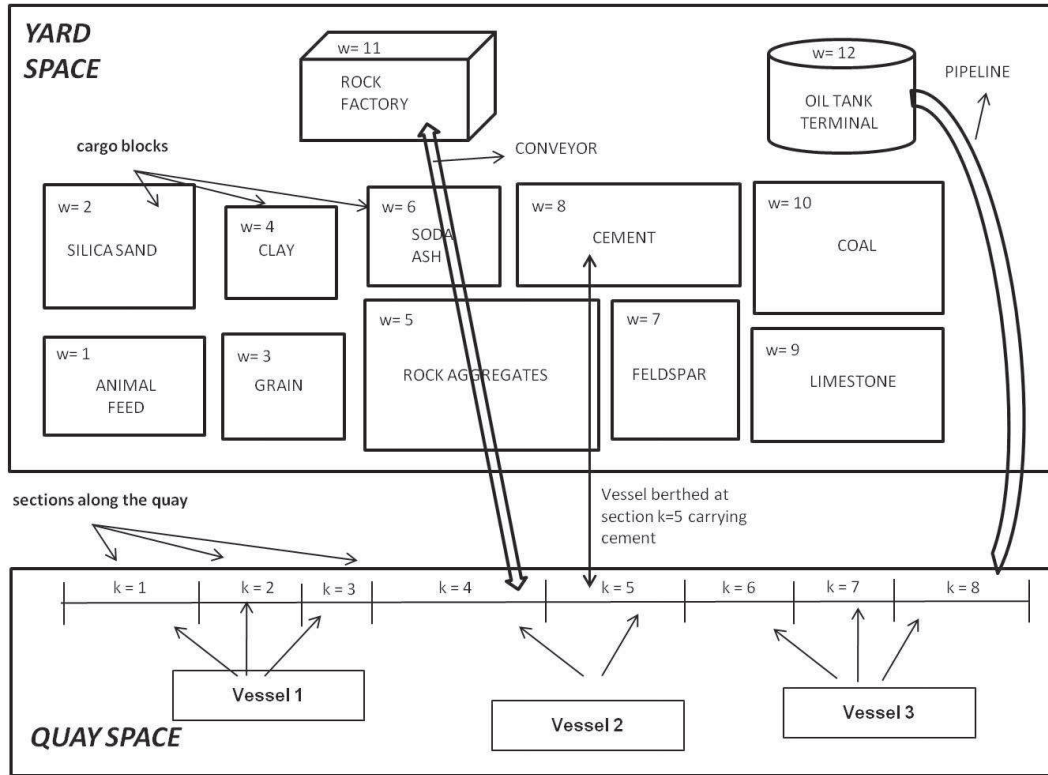


Figure 3.4: Schematic representation of a bulk port terminal

### 3.4 Models for BAP

In this section, we present two exact solution approaches and a heuristic approach to model the berth allocation problem in bulk ports with hybrid layout and dynamic vessel arrivals. In section 3.4.1, we describe a mixed integer linear programming approach to solve the problem, while in section 3.4.2, we use a set partitioning approach to model the same problem by apriori generating the set of all feasible berthing assignments by data pre-processing. In section 3.4.3, we describe a heuristic approach based on squeaky wheel optimization to obtain near-optimal solutions for large instances.

#### 3.4.1 MILP formulation

In this section, we present the MILP model for the dynamic, hybrid BAP in bulk ports. All temporal variables including the start time of operations and handling time of vessel, are modeled as continuous variables. The model uses several decision variables to obtain the berthing assignment of vessels to sec-

### Chapter 3. The Deterministic Berth Allocation Problem in Bulk Ports

---

tions along the quay as well as the berthing order of vessels at each section, as shown below:

- $m_i \geq 0$ , represents the starting time of handling of vessel  $i \in N$ ;
- $c_i \geq 0$ , represents the total handling time of vessel  $i \in N$ ;
- $s_k^i$  binary, equals 1 if section  $k \in M$  is the starting section of vessel  $i \in N$ , 0 otherwise;
- $x_{ik}$  binary, equals 1 if vessel  $i \in N$  occupies section  $k \in M$ , 0 otherwise;
- $y_{ij}$  binary, equals 1 if vessel  $i \in N$  is berthed to the left of vessel  $j \in M$  without any overlapping in space, 0 otherwise;
- $z_{ij}$  binary, equals 1 if handling of vessel  $i \in N$  finishes before the start of handling of vessel  $j \in N$ , 0 otherwise;

In the proposed formulation, we use additional parameters that are generated by data pre-processing that provide information on whether a particular section is occupied and the fraction of the length of the section that is occupied for a given vessel and given the first section occupied by the vessel. The following coefficients are generated and provided as input to the model:

$$d_{ilk} = \begin{cases} 1 & \text{if vessel } i \text{ starting at section } \ell \text{ touches section } k; \\ 0 & \text{otherwise.} \end{cases}$$

$$p_{ilk} = \begin{cases} \text{percentage of total cargo handled at section } k \text{ if vessel } i \text{ starts at section } \ell; \end{cases}$$

The MILP model for the dynamic berth allocation problem with hybrid berth layout in bulk ports is formulated as shown below.

$$\min \sum_i (m_i - A_i + c_i) \quad (3.4)$$

$$\text{s.t. } m_i - A_i \geq 0 \quad \forall i \in N \quad (3.5)$$

$$\sum_{k \in M} (s_k^j b_k) + B(1 - y_{ij}) \geq \sum_{k \in M} (s_k^i b_k) + L_i \quad \forall i, j \in N, i \neq j \quad (3.6)$$

$$m_j + B(1 - z_{ij}) \geq m_i + c_i \quad \forall i \in N, \forall j \in N, i \neq j \quad (3.7)$$

$$y_{ij} + y_{ji} + z_{ij} + z_{ji} \geq 1 \quad \forall i \in N, \forall j \in N, i \neq j \quad (3.8)$$

$$\sum_{k \in M} s_k^i = 1 \quad \forall i \in N \quad (3.9)$$

$$\sum_{k \in M} (s_k^i b_k) + L_i \leq L \quad \forall i \in N \quad (3.10)$$

$$\sum_{\ell \in M} (d_{i\ell} s_\ell^i) = x_{ik} \quad \forall i \in N, \forall k \in M \quad (3.11)$$

$$(d_k - D_i) x_{ik} \geq 0 \quad \forall i \in N, \forall k \in M \quad (3.12)$$

$$c_i \geq h_k^{W_i} p_{i\ell} Q_i s_\ell^i \quad \forall i \in N, \forall k \in M, \forall \ell \in M \quad (3.13)$$

$$s_k^i \in \{0, 1\} \quad \forall i \in N, \forall k \in M \quad (3.14)$$

$$x_{ik} \in \{0, 1\} \quad \forall i \in N, \forall k \in M \quad (3.15)$$

$$y_{ij} \in \{0, 1\} \quad \forall i, j \in N \quad (3.16)$$

$$z_{ij} \in \{0, 1\} \quad \forall i, j \in N \quad (3.17)$$

The objective function (3.4) minimizes the total service time of all vessels berthing at the port. Constraint (3.5) ensures that vessels can be serviced only after their arrival. Constraints (3.6)-(3.8) are the non-overlapping restrictions for any two vessels berthing at the port. Note that the constraints (3.6)-(3.7) have been linearized by using a large positive constant  $B$ . Constraints (3.9)-(3.11) ensure that each vessel occupies only as many number of sections as determined by its length and the starting section occupied by the vessel. Note that using a hybrid berthing layout in case of bulk ports is important. On the one hand, a discrete berthing layout will lead to inefficient allocation of vessels since the delays associated with the vessels requiring fixed specialized equipment facilities such as conveyors and pipelines will further increase. On the other hand, a continuous berthing layout introduces several non-linearities in the modeling of the handling times of vessels which are dependent on the berthing location of the vessels along the quay. Moreover in the hybrid berthing layout used in our

model, the sections are small enough such that a given vessel can occupy several sections while a given section can be occupied by at most one vessel at a given time. This maximizes the utilization of quay space ensuring better service quality of vessels requiring specialized equipment facilities such as conveyors and pipelines. Constraints (3.12) ensure that the draft of the vessel does not exceed the draft of any occupied section. Constraints (3.13) are used to determine the total handling time for any given vessel. The time taken to handle a given vessel at a given section is directly proportional to both the cargo quantity handled at that section as given by the product  $p_{ilk}Q_i$ , and the unit handling time associated with that section and cargo type given by the parameter  $h_k^w$ . The handling time at a section that is not occupied by the vessel is equal to zero as taken care of by the binary variable  $s_\ell^i$ , and the total handling time of the vessel is the handling time at the section with the maximum handling time value among all sections along the quay. Note that the above model is linear and thus small instances can be solved using general-purpose solvers.

The dynamic discrete BAP can be modeled as an unrelated machine scheduling problem (Pinedo (2002)), while in the continuous case the BAP can be modeled as a two dimensional cutting stock problem or bin packing problem (Lim (1998)). Thus in both cases, the problem is NP-hard (Garey and Johnson (1979)). The complexity of the proposed MILP based on hybrid berth layout is significantly increased because of the non-overlapping constraints (3.6)-(3.8), which contain several binary integer variables. This is because several combinations of specific integer values for the variables must be tested, and the number of such combinations rises exponentially with the size of the problem.

#### 3.4.2 GSPP formulation

The berth allocation problem described in Section 3.3 can also be modeled as a generalized set partitioning problem (GSPP). The GSPP model was proposed by Christensen and Holst (2008) in the context of container terminals. More recently, it has also been used by Buhrkal et al. (2011) to solve the discrete, dynamic BAP in container terminals. In the GSPP model, the planning horizon  $H$  is divided into discrete time intervals and only integral measurements of time are considered. The columns (variables) are generated apriori by data pre-processing, where a column represents a feasible berthing assignment of a single vessel to a specific set of section(s) at a specific time. In our research problem, the cargo type on the vessel is explicitly taken into account in the generation of feasible assignments for that particular vessel. For example, if a

Vessel 1	1	1	0	0
Vessel 2	0	0	1	1
Section 1, Time 1	1	0	0	0
Section 1, Time 2	1	1	1	0
Section 1, Time 3	0	1	1	0
Section 2, Time 1	1	0	0	0
Section 2, Time 2	1	1	1	1
Section 2, Time 3	0	1	1	1
Section 3, Time 1	0	0	0	0
Section 3, Time 2	0	0	0	1
Section 3, Time 3	0	0	0	1

Table 3.1: Assignment matrix for a simple example of GSPP

particular vessel carrying liquid bulk needs to be berthed at sections where the pipeline facility is installed, then only those assignments where the vessel is occupying these sections and the estimated departure time of the vessel does not exceed the length of planning horizon, are feasible. Alternatively, these vessels can be prevented from occupying sections that do not have the pipeline facility by providing extremely large handling time values associated with these sections and the cargo type corresponding to the conveyor facility. We illustrate the procedure with a small example containing two vessels 1 and 2. We consider 3 sections along the quay and 3 discrete time intervals in the planning horizon. Let us assume that vessel 1 needs the conveyor facility and cannot berth at section 3 which does not have this facility, while vessel 2 arrives at the start of time 2, and hence can only berth after that. Then the assignment matrix for the problem would look like as shown in Table 3.1. The first column represents the berthing assignment of vessel 1 to sections 1 and 2 from time 1-2, and so on.

We denote the set of columns by  $P$ . The assignment matrix is composed of the upper submatrix  $A$  and lower submatrix  $B$ . The upper submatrix  $A$  consists of  $|P|$  columns and  $N$  rows. In submatrix  $A$ , if column  $p \in P$  represents the feasible assignment of vessel  $i \in N$ , then the entry in row  $i$  is 1 while all other entries are zeroes. The lower submatrix  $B$  consists of  $|P|$  columns and a single row for every (section,time) position. Thus, in submatrix  $B$ , if column  $p \in P$ , represents the feasible assignment of vessel  $i \in N$ , then all entries corresponding to the (section, time) positions occupied by vessel  $i$  in the feasible assignment  $p \in P$  are 1, while all the remaining entries are zeroes.

We assume the following input data to be available for the GSPP model:

### Chapter 3. The Deterministic Berth Allocation Problem in Bulk Ports

---

$H =$  set of discrete time intervals in the planning horizon  
 $P =$  set of feasible assignments  
 $t = 1, \dots, |H|$  discrete time intervals in the planning horizon  
 $p = 1, \dots, |P|$  feasible assignments  
 $d_p =$  delay associated with assignment  $p$   
 $h_p =$  handling time associated with assignment  $p$

The assignment matrix coefficients are defined as follows.

$$A_{ip} = \begin{cases} 1 & \text{if the assigned vessel in feasible assignment } p \text{ is vessel } i; \\ 0 & \text{otherwise.} \end{cases}$$

$$b_p^{st} = \begin{cases} 1 & \text{if section } s \text{ is occupied at time } t \text{ in assignment } p; \\ 0 & \text{otherwise.} \end{cases}$$

There is only a single decision variable in the GSPP model for selection of feasible assignments in the optimal solution which is defined as follows.

$$\lambda_p = \begin{cases} 1 & \text{if assignment } p \text{ is part of the optimal solution;} \\ 0 & \text{otherwise.} \end{cases}$$

The GSPP model is formulated as shown below:

$$\min \sum_{p \in P} (d_p \lambda_p + h_p \lambda_p) \quad (3.18)$$

$$\text{s.t. } \sum_{p \in P} (A_{ip} \lambda_p) = 1 \quad \forall i \in N \quad (3.19)$$

$$\sum_{p \in P} (b_p^{kt} \lambda_p) \leq 1 \quad \forall k \in M, \forall t \in H \quad (3.20)$$

$$\lambda_p \in \{0, 1\} \quad \forall p \in P \quad (3.21)$$

Constraints (3.19) ensure that each vessel must have exactly one feasible assignment in the optimal solution. Constraints (3.20) ensure that a given section at a given time can be occupied by at most one vessel

GSPP is in general characterized as NP-hard. The growth in the number of variables and constraints in the set-partitioning approach is much faster as compared to the integer programming approach. Furthermore constraints involving two or more vessels are much more difficult to incorporate in the GSPP approach. For example in bulk context, it would be complicated to model con-

straints wherein two or more vessels cannot use the pipeline facilities installed at different sections at the same time. However, the approach offers several modeling advantages, primarily because it is much easier to incorporate more advanced spatial and temporal constraints on individual vessels as these can be easily handled while generating feasible assignments. It is also easier to model complex objectives as long as they can be expressed as a function of the column costs.

#### 3.4.3 SWO Heuristic

We now propose a meta-heuristic that improves the berthing assignment of vessels by iteratively changing the priority order of vessels with regards to the service quality measure of each vessel. The algorithm, commonly termed in literature as squeaky wheel optimization (SWO) works on the principle of Construct/ Analyze/ Prioritize, where the solution generated at each successive iteration is constructed and analyzed, and the results of this analysis are used to generate a new priority order to obtain the next solution. The algorithm operates on two search spaces: solutions and prioritizations as schematically shown in Figure 3.5. The idea of SWO was introduced by Clements et al. (1997) and has been used in several combinatorial optimization problems such as in scheduling problems and graph coloring problems, by Smith and Pyle (2004), Lim et al. (2004), Joslin and Clements (1998). In the context of container terminals, SWO algorithm has been used by few scholars such as Fu et al. (2007) to solve the port space allocation problem, and by Meisel and Bierwirth (2009) to solve the integrated berth allocation and crane assignment problem. The approach can be adapted to solve the dynamic, hybrid berth allocation problem in bulk ports as discussed in this section. SWO is typically useful in problems where it is possible to quantify the individual contribution of each single problem element to the overall solution quality. It could be used in our problem, since the objective is to minimize the total service time of the berthing schedule which is simply the sum of the service times of all vessels in the berthing schedule. Unlike in local search techniques such as iterated hillclimbing etc., the moves in search space are not motivated by the objective function value, but rather by the weak performing elements of the solution even when the move may lead to a worse overall solution.

In the implementation of the SWO algorithm in our problem, we use a base heuristic that returns a feasible berthing assignment for a given priority order of vessels. This is described by Algorithm 1. In the generation of feasible assign-



ments, the cargo type on the vessel is explicitly taken into account. The feasible assignments of vessels requiring fixed equipment facilities such as conveyors and pipelines have the vessels berthed at only those sections where these facilities are installed, or alternatively we can provide extremely large values of handling times for other sections along the quay. The initial solution is the berthing assignment obtained by prioritizing the incoming vessels in order of arrival times, also called the first-come-first-served (FCFS) ordering. At the end of each iteration, we assess the individual contribution of each vessel to the overall service time of the berthing assignment obtained, and rank the vessels according to their individual service performance to obtain a new priority order which is then used as an input in the next iteration. A vessel may be inserted between two vessels that have already been assigned or at the beginning or at the end of the berthing schedule. A given vessel at a particular rank in the given priority order is assigned to those set of sections which minimize the total waiting and handling time of the vessel, after all the vessels ranked above in the priority order have already been assigned.

---

**Algorithm 1:** Base heuristic in the implementation of SWO to solve BAP

---

**Require:** Priority list  $P$  of vessels, set  $M$  of sections

```
for  $i = 1 \rightarrow P$  do
  serviceTime = large constant  $B$ 
  startSection = large constant  $B$ 
  for  $k = 1 \rightarrow M$  do
    if IsStartSectionFeasible( $i, k$ ) then
      if GetServiceTime( $i, k$ )  $\leq$  serviceTime then
        serviceTime  $\leftarrow$  GetServiceTime( $i, k$ )
        startSection  $\leftarrow k$ 
      end if
    end if
  end for
  AssignVessel( $i$ , startSection)
end for
```

---

The key role of the prioritizer is to identify the weak performing vessels and move them forward in the sequence to enable them to be handled better by the constructor. Once these vessels start performing well, they sink back in the sequence at which point their performance may start deteriorating again and cause them to move forward in the sequence again. On the other hand, vessels performing consistently well sink back and stay there. In the bulk context, owing to the restrictive spatial constraints on vessels requiring specialized facilities such as conveyors and pipelines, the berthing delays for these vessels could be



---

**Algorithm 2:** Algorithm for implementation of SWO to solve BAP

---

**Require:** Set  $N$  of vessels, set  $M$  of sectionsObtain the initial priority list  $p_o$  given by increasing order of arrival timespriorityListsEvaluated =  $\{p_o\}$ current priority list  $p \leftarrow p_o$ **while** iteration  $\leq$  maxiterations **do**Obtain the berthing assignment  $b = \text{BaseHeuristic}(p)$ **for**  $i = 1 \rightarrow N$  **do**Calculate service quality  $s_i$  of vessel  $i$  in solution  $b$ **end for**Sort the vessels in decreasing order of  $s_i$  to obtain the updated priority list $p$ **while** priorityListsEvaluated contains  $p$  **do** $p = \text{Randomize}(p)$ **end while**Add  $p$  to priorityListsEvaluated**if** service time( $b$ )  $\leq$  service time(finalBerthingAssignment) **then**finalBerthingAssignment  $\leftarrow b$ **end if****end while**

---

very large and these vessels are typically the weakest performing elements in the overall solution. Such vessels are heavily penalized if they are ranked lower in the priority list, and thus to ensure high service quality for these vessels in particular, they should not be allowed to sink back in the priority list. In the SWO algorithm, there is also the risk of the algorithm getting trapped in a cycle alternating between a set of priority listings. To avoid this, if a listing has already been evaluated, we generate a new listing by introducing some randomization in the current priority order by swapping two or more vessels until we get a priority listing that has not been evaluated so far. The algorithm terminates after a preset number of iterations, and the best solution obtained thus far is accepted as the final solution. The implementation is described by Algorithm 2.

One key issue in using SWO for large problem size is that a new solution is constructed from scratch after every iteration. A possible solution to this problem could be the use of a history mechanism that keeps track of the previous solutions generated for given prioritizations, as that would speed up the construction process. Another bottleneck in the approach is that in many cases, the optimal solution has some badly performing vessels. This is a problem since

this approach is primarily motivated by identifying such vessels and assigning high blame to them to move them forward in the sequence and enable them to be handled better in the next iteration. This prevents the SWO approach to identify such sacrificial vessels and converge to good solutions. While a deeper understanding of the approach would definitely help to obtain better solutions, in this study we have shown that a relatively simple implementation of the approach can be used to obtain reasonably good results for our problem.

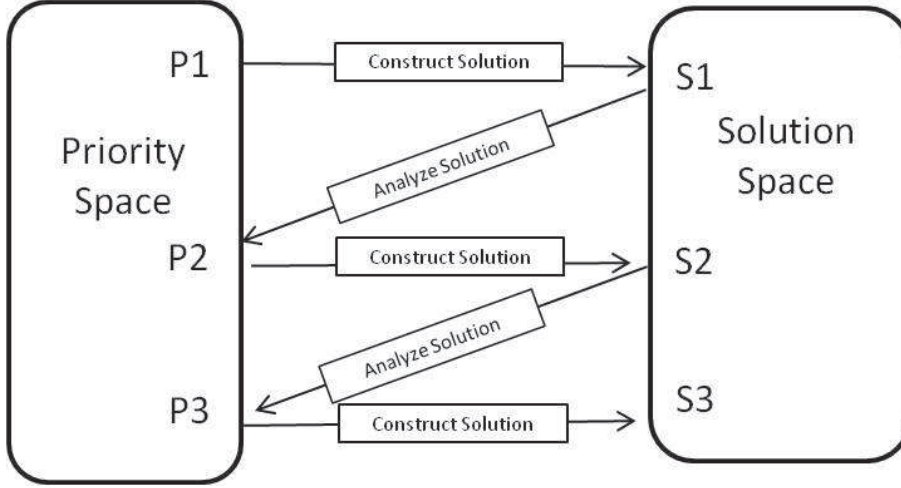


Figure 3.5: Schematic Representation of SWO Algorithm

## 3.5 Results and Analysis

In this section, we compare the different BAP formulations presented in earlier sections. The MILP formulation described in Section 3.4.1 is tested using CPLEX solver with the solution time limit set to 7200 seconds. In the GSPP approach, all feasible assignments for the given instances are generated a priori using JAVA code and provided as an input to the GSPP formulation described in Section 3.4.2. The optimization model in (3.18)- (3.21) is then solved using CPLEX solver. The heuristic algorithm presented in Section 3.4.3 is implemented in JAVA programming language. All tests were run on an Intel Core i7 (2.80 GHz) processor and used a 32-bit version of CPLEX 12.2.

### 3.5.1 Generation of Instances

The instances were generated based on a small sample of data received from SAQR port, Ras Al Khaimah (RAK), UAE. SAQR port is the biggest bulk com-

modity port in the entire middle east handling approximately 30 million tonnes of cargo annually. The port plays a key role in the economic growth of the RAK emirate, which has registered a significant growth in GDP from AED 6.6 billion in 2002 to AED 13.6 billion in 2008. The port's cargo handling department specializes in dealing with a wide variety of imported and exported commodities including consignments of aggregates, cement, coal, clinker, iron ore, feldspar, clay, soda ash, silica sand, grain, animal feedstock, steel, project cargoes and petroleum products (www.saqrport.com).

The data sample received from the port provided information about the vessel lengths, expected and actual times of arrival, berthing, processing and departure of vessels, expected and actual berthing positions and the cargo tonnage of the vessels. The data was provided for over 20 vessels for a time horizon of roughly 10 days from 28th March to 6th April, 2011. Although, there were a lot of missing entries in the data file, we could use the data sample to get a rough estimate of the range of values for most input parameters in our model.

To do a comparison of the different formulations, we generate 6 instance sizes with  $|N| = 10, 25$  and  $40$  vessels and  $|M| = 10$  and  $30$  sections along the quay. A set of 9 instances was generated for each instance size. In all instances, the total quay length  $L$  is 1600 meters, and the vessel lengths  $L_i$  lie in the range 80-260 meters as in SAQR. The expected arrival times  $A_i$  are randomly generated between a given range of values, described more explicitly later in the chapter. We further remark that the drafts of all vessels  $D_i$  are less than the minimum draft for all sections, as in the data provided by the port. Therefore, constraints (3.12) are never active for the tested instances.

In the generation of handling times  $h_k^w$ , we consider six cargo types - clay, grain, coal, cement, conveyor and pipeline. The value of the parameter  $T$  is based on the crane handling rate of 1000 tonnes per hour. The number of cranes operating in section  $k$  is determined by the length of the section, assuming an additional crane for every 50 meters of section length. In Figure 3.6, the cargo locations on the yard with respect to the quay axis have been graphically shown. The distance  $d_k^w$  in equation 3.3 is calculated as the Euclidean distance between the midpoint of section  $k$  and the cargo location  $w$ . The parameter  $v^w$  which is dependent on the rate of transfer of cargo is assumed to be equal to 1/1200 hours per meter per unit cargo for the conveyor, 1/3600 hours per meter per unit cargo for the pipeline and 1/600 hours per meter per unit cargo for all other cargo types.

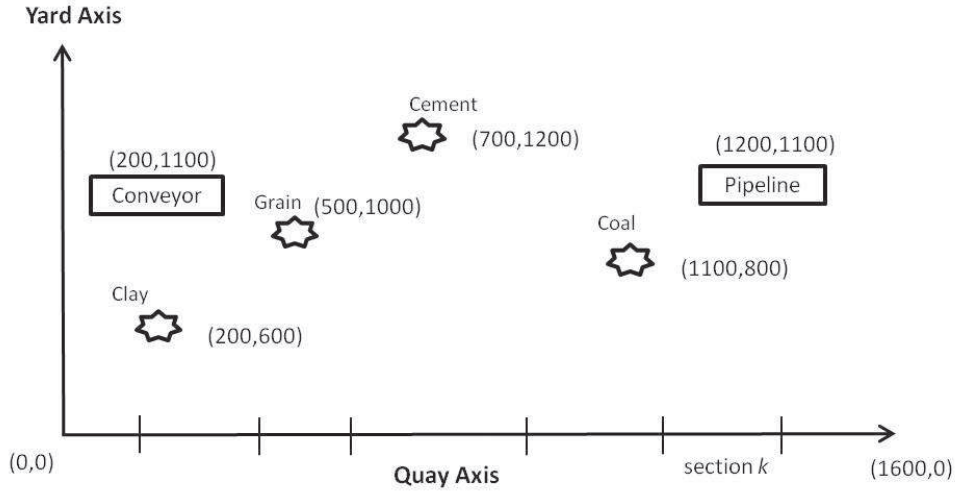


Figure 3.6: Graphical Representation of the yard layout used in the generation of handling times

The modeling results were found to be sensitive to the discretization used in the berthing layout, even for the same number of sections. In the instances shown, the berthing layouts have been fixed for  $|M| = 10, 30$ . The minimum section length is 25 meters, and all section lengths are whole number multiples of this minimum section length. We use the berthing layouts as shown in Table 3.2 and Table 3.3.

To test the sensitivity of the results with respect to the different input parameters in the model, the instances have been designed as shown in Table 3.4.

Section	Length	Facility
1	150	C
2	50	C
3	200	P
4	150	C,P
5	125	C,P
6	250	-
7	250	-
8	75	P
9	150	P
10	200	-

Table 3.2: Berthing layout and fixed facility locations for  $|M|=10$  (C and P stand for conveyor and pipeline respectively)

Section	Length	Facility
1	150	C
2	25	C
3	25	C
4	25	P
5	25	P
6	100	P
7	25	P
8	25	P
9	25	C,P
10	25	C,P
11	100	C,P
12	25	C,P
13	25	C,P
14	25	C,P
15	50	C,P
16	50	-
17	75	-
18	200	-
19	75	-
20	50	-
21	50	-
22	25	P
23	25	P
24	25	P
25	100	P
26	25	P
27	25	P
28	25	-
29	25	-
30	150	-

Table 3.3: Berthing layout and fixed facility locations for  $|M|=30$  (C and P stand for conveyor and pipeline respectively)

Instance	Vessel Lengths	Congestion	Instance	Vessel Lengths	Congestion
N =10,  M =10			N =10,  M =30		
A1	80-260 m	yes	B1	80-260 m	yes
A2	80-170 m	yes	B2	80-170 m	yes
A3	170-260 m	yes	B3	170-260 m	yes
A4	80-260 m	mild	B4	80-260 m	mild
A5	80-170 m	mild	B5	80-170 m	mild
A6	170-260 m	mild	B6	170-260 m	mild
A7	80-260 m	no	B7	80-260 m	no
A8	80-170 m	no	B8	80-170 m	no
A9	170-260 m	no	B9	170-260 m	no
N =25,  M =10			N =25,  M =30		
C1	80-260 m	yes	D1	80-260 m	yes
C2	80-170 m	yes	D2	80-170 m	yes
C3	170-260 m	yes	D3	170-260 m	yes
C4	80-260 m	mild	D4	80-260 m	mild
C5	80-170 m	mild	D5	80-170 m	mild
C6	170-260 m	mild	D6	170-260 m	mild
C7	80-260 m	no	D7	80-260 m	no
C8	80-170 m	no	D8	80-170 m	no
C9	170-260 m	no	D9	170-260 m	no
N =40,  M =10			N =40,  M =30		
E1	80-260 m	yes	F1	80-260 m	yes
E2	80-170 m	yes	F2	80-170 m	yes
E3	170-260 m	yes	F3	170-260 m	yes
E4	80-260 m	mild	F4	80-260 m	mild
E5	80-170 m	mild	F5	80-170 m	mild
E6	170-260 m	mild	F6	170-260 m	mild
E7	80-260 m	no	F7	80-260 m	no
E8	80-170 m	no	F8	80-170 m	no
E9	170-260 m	no	F9	170-260 m	no

Table 3.4: Description of Instances

### 3.5.2 Comparison of Algorithms

The computational results for the three approaches presented earlier are shown in Tables 3.5-3.6. Results obtained from the first-come-first-served (FCFS) heuristic which is used to obtain an initial solution in the implementation of the SWO heuristic algorithm have been also shown.

As can be seen from the results, for  $|N| = 10$  vessels, the MILP formulation produces optimal results for all instances within the prescribed CPLEX time limit of 2 hours. However, it is not able to solve even a single instance to optimality for instances with  $|N| = 25, 40$ , with a significantly large duality gap at the end of the run. For instance F3 which represents the congested scenario with  $|N| = 40$  vessels and  $|M| = 30$  sections, the model is unable to find even a single feasible integer solution within the CPLEX time limit. Clearly, the complexity of the problem is highly affected by the problem size and increases significantly, which as discussed earlier can be attributed to the exponentially increasing number of integer variables with increase in problem size.

In the GSPP formulation, we generate feasible assignments for a sufficiently large planning horizon of 150 hours, divided into discrete time intervals of 1 hour. It should be noted that the computational time provided for GSPP model includes the time taken to generate the feasible assignments and subsequently solve the optimization model using CPLEX. As can be seen from the results, the performance of the GSPP model is quite remarkable, as it is able to solve all instances to optimality, and most of them within few minutes of computational time. It may however be noted that the GSPP does not reach the same objective as the MILP formulation, even though it reaches optimality. This is because of the fact that we consider discrete time intervals in the GSPP method, while time is modeled as continuous in the MILP formulation. For instance id F with  $|N| = 40$  and  $|M| = 30$ , the GSPP model runs out of memory when the length of time interval  $h$  is equal to 1 hour, since the number of feasible assignments is very large. To overcome this problem, we use time intervals of  $h=2$  hours for instance id F.

The FCFS heuristic produces results by simply berthing vessels according to their arrival order. The heuristic used to obtain an initial solution in the implementation of the SWO, performs reasonably well for small sized instances but the performance is weak for larger instances. This indicates that as instance size grows, the berthing order has a larger deviation from the arrival order of vessels.



The SWO heuristic performs reasonably well for the tested instances. The optimality gap is less than 10 percent averaged over all the tested 54 instances. Since the feasible assignments are explicitly enumerated in the GSPP approach, it runs the risk of running out of memory for a large number of assignments. Even when the time interval  $h$  is equal to 2 hours, the GSPP formulation takes a relatively larger time to converge for some instances, as can be specially seen for instances F3, F6 and F9 which represent the instances with large vessels with  $|N| = 40$  vessels and  $|M| = 30$  sections. But the SWO heuristic provides near optimal solutions in much less time as compared to GSPP. Thus, for very large instances or even smaller sized instances with long planning horizons, SWO may be used to obtain sub-optimal berthing assignments.

Table 3.5: Computational results for generated instances, Instances A, B, C

Instance	MILP			GSPP			FCFS			SWO		
	obj	gap <sup>a</sup>	time <sup>b</sup>	obj	time( H =150, t=1) <sup>c</sup>	obj	obj	RE <sup>d</sup>	obj	obj	RE <sup>d</sup>	time
10x10												
A1	230.21	0.01%	67.67	231.21	5.94	262.09	262.09	13.36%	230.48	230.48	-0.32%	15.81
A2	237.35	0.01%	15.31	238.49	5.54	250.44	250.44	5.01%	239.08	239.08	0.25%	16.66
A3	223.99	0.01%	9.58	226.61	5.96	280.04	280.04	23.58%	225.33	225.33	-0.57%	16.97
A4	227.12	0.01%	10.31	227.22	5.68	240.91	240.91	6.03%	228.00	228.00	0.35%	16.60
A5	234.20	0.01%	5.60	234.22	5.43	251.09	251.09	7.20%	234.47	234.47	0.11%	16.30
A6	233.12	0.01%	11.06	234.06	6.85	262.61	262.61	12.20%	233.12	233.12	-0.40%	16.90
A7	203.23	0.00%	0.56	203.23	4.99	208.44	208.44	2.57%	203.38	203.38	0.07%	15.95
A8	218.87	0.00%	0.56	219.99	5.29	220.90	220.90	0.41%	218.87	218.87	-0.51%	16.72
A9	198.03	0.00%	0.60	199.89	5.16	214.17	214.17	7.14%	198.03	198.03	-0.93%	17.53
Mean								8.61%			-0.22%	
10x30												
B1	188.39	0.01%	15.80	189.73	94.552	216.56	216.56	14.14%	192.81	192.81	1.62%	49.95
B2	178.07	0.01%	15.78	179.10	86.08	186.41	186.41	4.08%	178.08	178.08	-0.57%	48.42
B3	200.16	0.01%	1094.61	202.33	101.93	230.14	230.14	13.74%	216.14	216.14	6.82%	49.79
B4	182.57	0.01%	3.04	184.27	89.58	224.00	224.00	21.56%	182.80	182.80	-0.80%	47.73
B5	178.48	0.01%	10.97	179.23	85.01	185.60	185.60	3.55%	179.01	179.01	-0.12%	48.37
B6	199.82	0.01%	87.78	201.17	96.19	240.09	240.09	19.35%	223.37	223.37	11.04%	48.83
B7	173.02	0.01%	1.30	173.02	86.00	175.72	175.72	1.56%	175.30	175.30	1.32%	48.61
B8	162.51	0.00%	1.57	162.81	81.67	169.20	169.20	3.92%	166.20	166.20	2.08%	50.29
B9	175.29	0.00%	1.39	175.81	95.74	192.41	192.41	9.44%	191.26	191.26	8.79%	50.27
Mean								10.15%			3.35%	
25x10												
C1	812.32	33.08%	-	819.22	14.09	976.49	976.49	19.20%	869.31	869.31	6.11%	22.29
C2	783.45	30.27%	-	781.72	11.70	924.35	924.35	18.25%	825.92	825.92	5.65%	22.58
C3	903.51	33.39%	-	900.43	20.19	1107.59	1107.59	23.01%	929.32	929.32	3.21%	22.24
C4	795.71	23.18%	-	791.18	15.26	877.90	877.90	10.96%	852.03	852.03	7.69%	23.28
C5	751.19	27.47%	-	747.88	10.41	846.26	846.26	13.15%	774.17	774.17	3.51%	22.16
C6	874.53	24.50%	-	863.86	19.38	979.26	979.26	13.36%	898.44	898.44	4.00%	23.19
C7	735.13	19.72%	-	741.16	15.91	840.85	840.85	13.45%	806.23	806.23	8.78%	22.54
C8	689.37	22.26%	-	699.14	11.23	761.48	761.48	8.92%	735.46	735.46	5.20%	22.32
C9	800.00	22.76%	-	793.24	12.82	936.55	936.55	18.07%	872.76	872.76	10.03%	23.56
Mean								15.37%			6.02%	

<sup>a</sup> The gap is calculated with respect to the value of linear relaxation as (Upper bound - LP lower bound) / Upper bound.<sup>b</sup> "-" indicates that the CPLEX time limit of 2 hours was reached. "+" indicates that CPLEX was unable to find a feasible integer solution within the time limit.<sup>c</sup> "\*" indicates that the CPLEX solver ran out of memory for t=1 hour, and t = 2 hours for these instances to reduce number of feasible assignments.<sup>d</sup> The relative error, RE is calculated with respect to the objective function value of GSPP model.

Table 3.6: Computational results for generated instances, Instances D, E, F

Instance	MILP			GSPP			FCFS			SWO		
	obj	gap <sup>a</sup>	time <sup>b</sup>	obj	time( H =150, t=1) <sup>c</sup>	obj	obj	RE <sup>d</sup>	obj	obj	RE <sup>d</sup>	time
25x30												
D1	690.79	23.14%	-	670.42	219.04	857.99	857.99	27.98%	785.91	785.91	17.23%	105.36
D2	617.31	34.23%	-	591.06	185.44	723.13	723.13	22.34%	667.20	667.20	12.88%	96.03
D3	809.55	30.75%	-	784.94	387.40	984.28	984.28	25.40%	866.82	866.82	10.43%	105.66
D4	657.48	20.62%	-	636.19	220.40	778.50	778.50	22.37%	728.60	728.60	14.53%	100.95
D5	560.65	25.96%	-	556.37	172.09	622.14	622.14	11.82%	614.72	614.72	10.49%	91.35
D6	754.87	21.97%	-	739.44	253.67	909.53	909.53	23.00%	836.23	836.23	13.09%	102.34
D7	581.54	8.36%	-	590.24	194.80	731.55	731.55	23.94%	706.82	706.82	19.75%	100.82
D8	510.80	16.20%	-	506.30	167.09	565.87	565.87	11.77%	565.87	565.87	11.77%	111.46
D9	704.76	19.18%	-	677.97	200.63	848.97	848.97	25.22%	778.67	778.67	14.85%	99.87
Mean								21.54%			13.89%	
40x10												
E1	1243.64	63.77%	-	1140.60	41.73	1536.78	1536.78	34.73%	1289.88	1289.88	13.09%	28.24
E2	1193.05	59.69%	-	1138.16	18.80	1571.07	1571.07	38.04%	1273.09	1273.09	11.86%	28.44
E3	1341.65	67.35%	-	1249.06	139.47	1878.78	1878.78	50.42%	1416.54	1416.54	13.41%	31.50
E4	1113.36	59.53%	-	1051.50	30.87	1408.95	1408.95	33.99%	1137.20	1137.20	8.15%	30.11
E5	1105.34	56.98%	-	1063.85	19.06	1447.39	1447.39	36.05%	1202.50	1202.50	13.03%	32.06
E6	1361.62	68.15%	-	1160.05	167.58	1903.39	1903.39	64.08%	1330.64	1330.64	14.71%	34.01
E7	1011.20	55.47%	-	946.35	26.04	1291.11	1291.11	36.43%	1148.14	1148.14	21.32%	31.69
E8	1013.41	53.02%	-	953.24	20.03	1183.57	1183.57	24.16%	1094.15	1094.15	14.78%	32.01
E9	1181.97	64.95%	-	1071.46	94.88	1500.71	1500.71	40.06%	1296.79	1296.79	21.03%	35.64
Mean								39.77%			14.60%	
40x30												
F1	1193.42	70.56%	-	920.73	506.02*	1278.04	1278.04	38.81%	1092.44	1092.44	18.65%	169.53
F2	913.59	62.66%	-	863.43	127.91*	1168.60	1168.60	35.34%	911.47	911.47	5.56%	159.28
F3	+	+	+	1089.48	932.56*	1784.98	1784.98	63.84%	1411.24	1411.24	29.53%	173.89
F4	902.74	59.15%	-	856.41	3341.62*	1143.59	1143.59	33.53%	1035.72	1035.72	20.94%	160.113
F5	881.37	61.20%	-	786.27	137.91*	973.94	973.94	23.87%	857.47	857.47	9.06%	163.41
F6	1121.14	66.39%	-	1015.53	2281.78*	1628.76	1628.76	60.39%	1286.73	1286.73	26.71%	265.29
F7	922.04	62.05%	-	777.06	829.88*	932.34	932.34	19.98%	932.34	932.34	19.98%	166.52
F8	728.48	52.93%	-	679.58	131.81*	774.12	774.12	13.91%	745.00	745.00	9.63%	160.49
F9	934.35	58.59%	-	920.29	1767.57*	1458.45	1458.45	58.48%	1214.66	1214.66	31.99%	171.97
Mean								38.68%			19.12%	

<sup>a</sup> The gap is calculated with respect to the value of linear relaxation as (Upper bound - LP lower bound) / Upper bound.<sup>b</sup> "-" indicates that the CPLEX time limit of 2 hours was reached. "+" indicates that CPLEX was unable to find a feasible integer solution within the time limit.<sup>c</sup> "\*" indicates that the CPLEX solver ran out of memory for t=1 hour, and t = 2 hours for these instances to reduce number of feasible assignments.<sup>d</sup> The relative error, RE is calculated with respect to the objective function value of GSPP model.

### 3.5.3 Results Analysis

In Figures 3.7-3.8, the optimal solutions of the instances A1 and C1 are graphically represented on the time space diagram as shown. Each rectangle represents a vessel whose width represents the berthing location of the vessel along the quay, and the height represents the handling time of the vessel. The cargo type on each vessel is marked on the rectangle representing that particular vessel. The fixed facility locations are marked along the quay axis. It can be seen that the optimal berthing locations for most vessels are at close physical proximity to the location of the vessel cargo type on the yard, as per the yard layout shown in Figure 3.6. This is the case since the distance between the berthing location of the vessel and the yard location of the cargo type of the vessel is explicitly considered in modeling the handling times of the vessels. The vessels requiring specialized equipment facilities are berthed at sections where these facilities are installed by providing appropriate values of handling times for each combination of section and cargo type. Thus for a given yard layout of the bulk terminal our model ensures better coordination between the berthing and yard activities, apart from minimizing the total service time of all vessels berthing at the port. It may be noted however that some vessels requiring the fixed equipment facilities experience large delays owing to the restrictive spatial constraints on these vessels.

The computational results obtained from the GSPP formulation are used to investigate the impact of vessel lengths, berthing layouts and congestion on the optimal service times and complexity of the BAP. In Figure 3.11, the percentage difference in optimal service time values for  $|M|=10$  and  $|M|=30$  are plotted for different number of vessels. It is clear from the plots that having more sections leads to better service times owing to better utilization of the quay space. It is interesting to note however that the difference in optimal service times is most significant for instances with smaller vessels represented by indices (2,5,8). This clearly indicates that instances with small vessels are more sensitive to the berthing layouts and choosing a higher number of sections is more advantageous for such instances. As can be seen in Table 3.6, in many cases for  $|N|=40$  vessels, the computational time is significantly higher for  $|M|=30$  sections as compared to  $|M|=10$  sections. Thus, in choosing the discretization for larger problem size, there may be a trade-off between obtaining a better solution value and obtaining the solution in reasonable computational time.

In Figure 3.12, the optimal service times have been plotted for each instance size with varying degrees of congestion for vessel lengths in the range 80-260

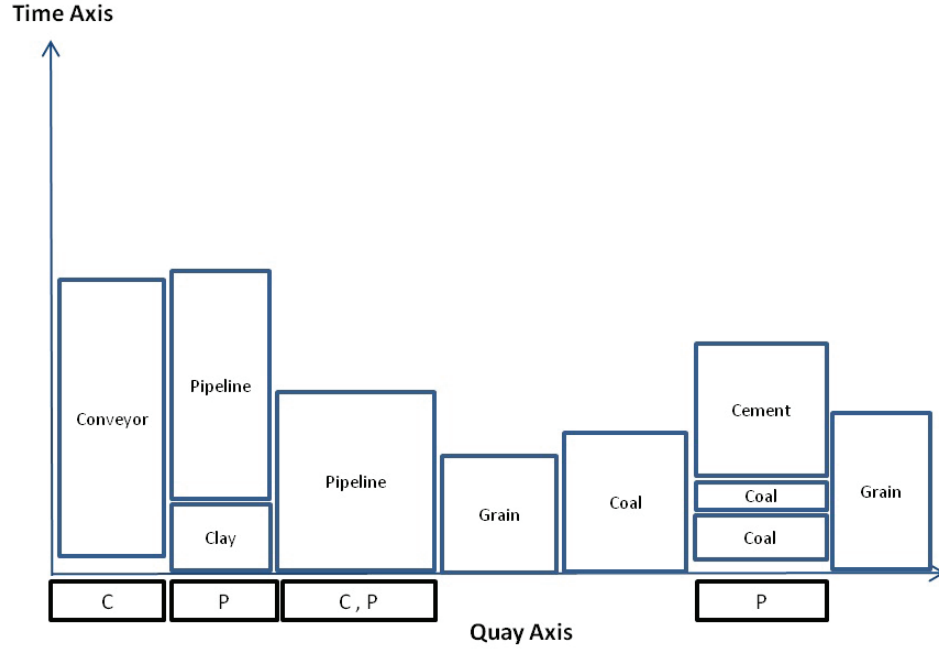
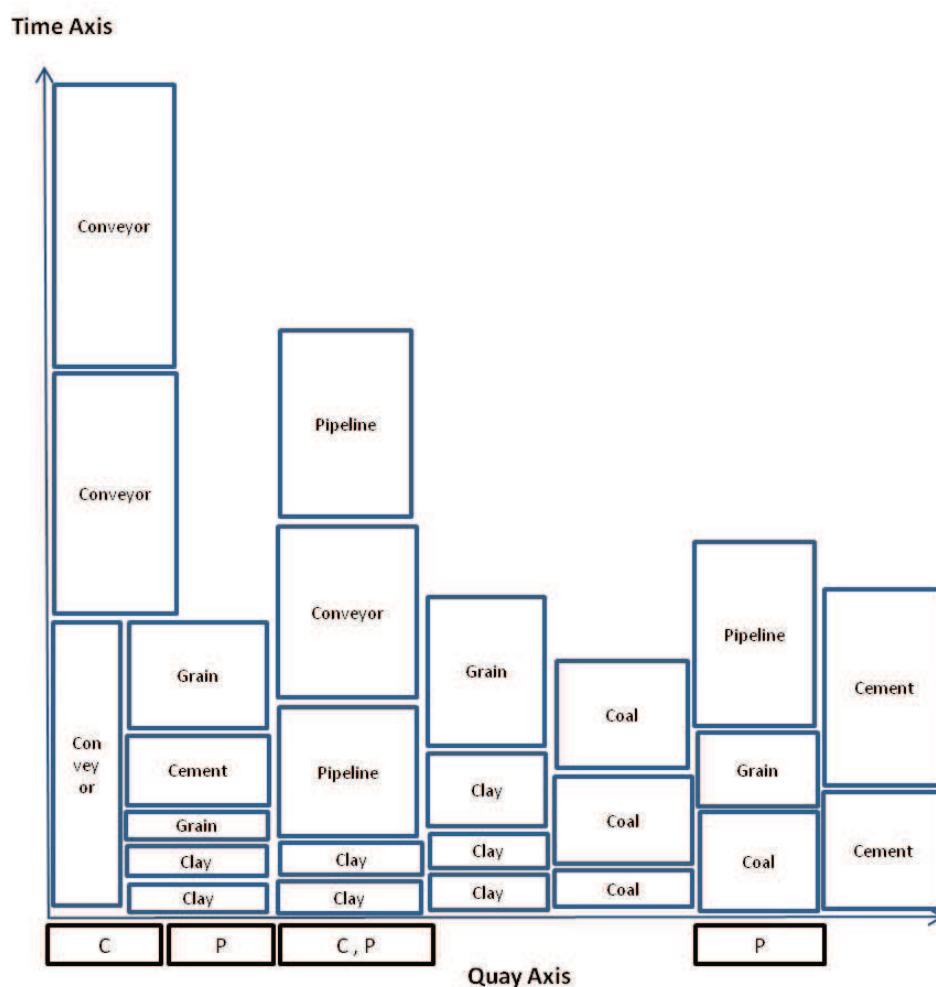


Figure 3.7: Graphical Representation of the optimal solution output for the instance A1. The fixed facility locations are marked along the quay, where C and P stand for conveyor and pipeline respectively.

meters represented by instance ids (1,4,7). Here, congestion is used to represent the case when vessel arrivals are very close together in time, as opposed to the congestion free case when arrivals are widely spaced in time. It can be seen from the plot, that as instance size grows the effect of congestion is also higher as indicated by the negative slopes of the curves for the larger instances. Thus it can be inferred that the temporal proximity of vessel arrivals enhances the complexity of the problem, and leads to higher service time values.

### 3.6 Conclusions and Future Directions

In this research, we have presented and compared three different formulations to solve the dynamic, hybrid BAP in bulk port terminals. The contributions of this research to the existing literature in port operations planning are as follows. The primary contribution is to model and solve the berth allocation problem in the context of bulk ports, which have received no attention in the literature thus far. The key difference from container terminals is that in bulk ports it is necessary to explicitly account for the cargo type on the vessel and the fixed equipment facilities such as conveyors and pipelines which are installed at only



model outperforms other MIP formulations and meta-heuristics for the discrete berth allocation problem considered in their paper. For our problem, the GSPP model was able to solve all the tested instances to optimality. However, the number of variables and constraints in the GSPP model grows very fast with the increase of the instance size.

A third contribution is to add bulk-specific components to the metaheuristic based on squeaky wheel optimization previously used by Fu et al. (2007) and Meisel and Bierwirth (2009) in the context of container terminals. For our problem, the modified heuristic produces sub-optimal results with less than 10 percent gap averaged over all the tested instances with respect to the GSPP solution. Thus it could be used as an alternative in cases where GSPP model is too slow or does not provide any result.

The presented models and methodologies show promising results for berth allocation planning in bulk ports. However, many aspects are worth being investigated in future research. To consider multiple cargo types on the same vessel, the internal structure of the vessels needs to be modeled more rigorously. On the methodological front, we believe that the proposed set-partitioning method can be used to solve very large sized instances to optimality using a branch-and-price framework in which the feasible assignments are generated and added to the active pool of columns in a dynamic way, instead of solving the problem by apriori generating the set of all feasible assignments. Another challenging problem is to make the proposed model more robust to account for unforeseen disruptions in operations owing to uncertainties in arrival times and handling times of vessels, and the breakdown of specialized facilities such as conveyors or pipelines which is more specific to bulk ports.



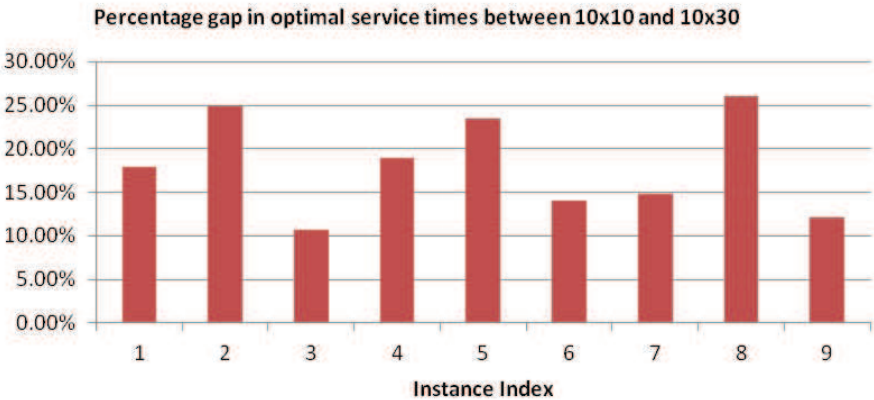


Figure 3.9: Effect of discretization and vessel lengths on service times for  $|N| = 10$  vessels

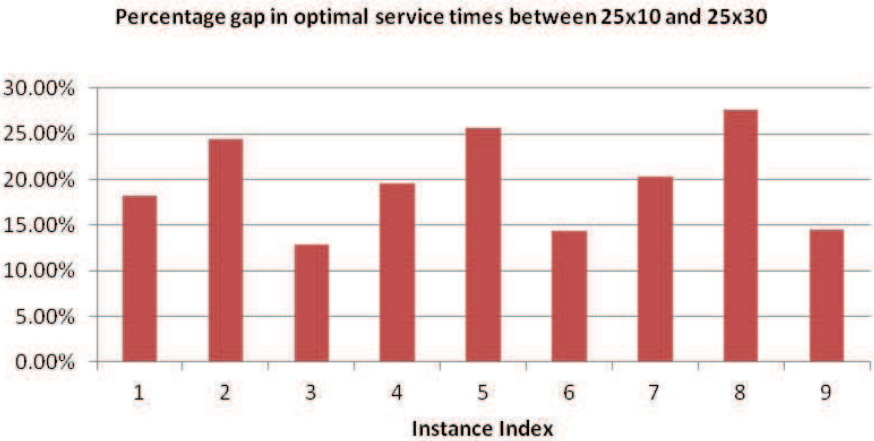


Figure 3.10: Effect of discretization and vessel lengths on service times for  $|N| = 25$  vessels

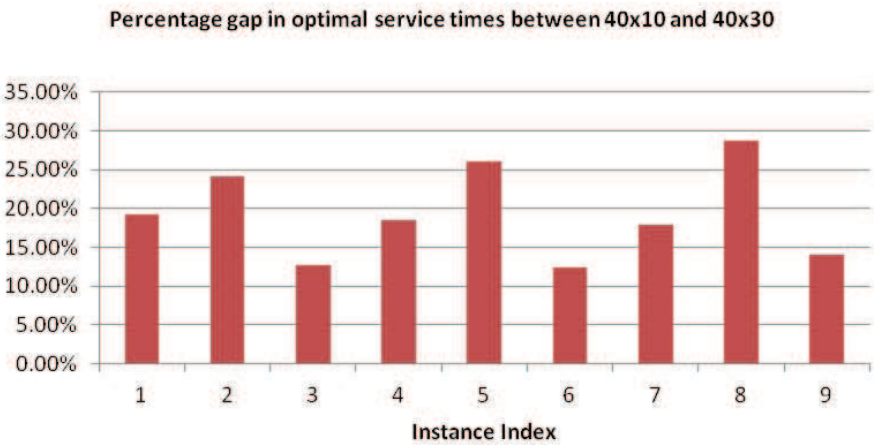


Figure 3.11: Effect of discretization and vessel lengths on service times for  $|N| = 40$  vessels



### 3.6. Conclusions and Future Directions

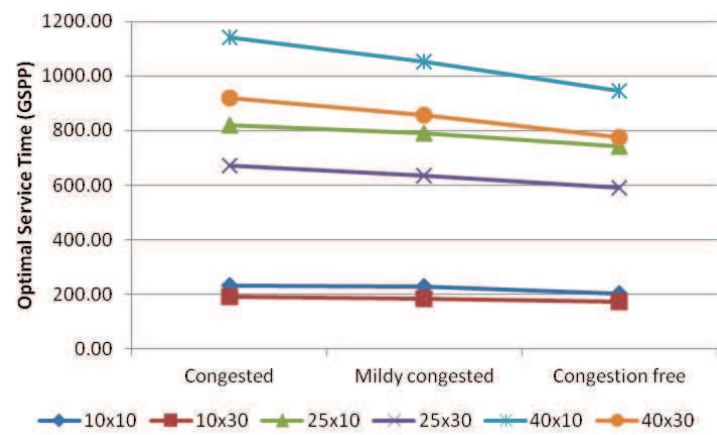


Figure 3.12: *Effect of congestion on service times*



## 4 The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports

In this chapter, two crucial optimization problems of berth allocation and yard assignment in the context of bulk ports are studied. We discuss how these problems are interrelated and can be combined and solved as a single large scale optimization problem. More importantly we highlight the differences in operations between bulk ports and container terminals which highlights the need to devise specific solutions for bulk ports. The objective is to minimize the total service time of vessels berthing at the port. We propose an exact solution algorithm based on a branch and price framework to solve the integrated problem. In the proposed model, the master problem is formulated as a set-partitioning problem, and the sub-problems to identify columns with negative reduced costs are solved using mixed integer programming. To obtain sub-optimal solutions quickly, a meta-heuristic approach based on critical-shaking neighborhood search is presented. The proposed algorithms are tested and validated through numerical experiments based on instances inspired from real bulk port data. The results indicate that the algorithms can be successfully used to solve instances containing up to 40 vessels within reasonable computational time.

### 4.1 Introduction

Bulk terminal operations planning can be divided into two decision levels depending on the time frame of decisions: Tactical Level and Operational Level. Tactical level decisions involve medium to short term decisions regarding resource allocation such as port equipment and labor, berth and yard management, storage policies etc. In practice, these decisions could be based on "rules of thumb" in which the experience of the port managers plays an important

## **Chapter 4. The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports**

---

role, or alternatively more scientific approaches based on operations research methods could be in use. The operational level involves making daily and real time decisions such as crane scheduling, yard equipment deployment and last minute changes in response to disruptions in the existing schedule. This chapter focuses on the tactical level decision planning for the integrated berth and yard management in the context of bulk ports. We focus in particular on two crucial optimization problems in the context of bulk port terminals: The Berth Allocation Problem (BAP) and the Yard Assignment Problem.

The tactical berth allocation problem refers to the problem of assigning a set of vessels to a given berthing layout within a given time horizon. There could be several objectives such as the minimization of the service times of vessels, the minimization of the port stay time, the minimization of the number of rejected vessels, the minimization of the deviation between actual and planned berthing schedules etc. There are several spatial and temporal constraints involved in the BAP, which lead to a multitude of BAP formulations. The temporal attributes include the vessel arrival process, the start of service, the handling times of vessels, while the spatial attributes relate to the berth layout, the draft restrictions and others. In a container terminal, all cargo is packed into containers, and thus there is no need for any specialized equipment to handle any particular type of cargo. In contrast, in bulk ports, depending on the vessel requirements and cargo properties, a wide variety of equipment is used for discharging or loading operations. Thus, the cargo type on the vessel needs to be explicitly taken into consideration while modeling the berth allocation problem in bulk ports.

The tactical yard assignment problem refers to decisions that concern the storage location and the routing of materials. This affects the travel distance between the assigned berth to the vessel and storage location of the cargo type of the vessel on the yard, and furthermore determines the storage efficiency of the yard. Thus, the problems of berth allocation and yard management are inter-related. The start and end times of vessel operations determine the workload distribution and the deployment of yard equipment such as loading shovels and wheel loaders in the yard side. Moreover, berthing locations of vessels determine the storage locations of specific cargo types to specific yard locations, which minimize the total travel distance between the assigned berthing positions to the vessels and the yard locations storing the cargo type for the vessel. Similarly, the yard assignment of specific cargo types has an impact on the best berthing assignment for vessels berthing at the port. In this study, we present an integrated model for the dynamic, integrated berth allocation problem and yard assignment in the context of bulk ports. Few scholars have inves-

tigated this problem in the context of container terminals, and there is no published literature for bulk ports. We present an exact solution algorithm based on branch-and-price and a metaheuristic approach based on critical-shaking neighborhood search to solve the combined large scale problem. Numerical experiments based on real-life-inspired port data indicate that the proposed algorithms can be successfully used to solve even large instances.

## 4.2 Literature Review

From the past OR literature on container terminal operations, it is well established that integrated planning of operations can allow port terminals to reduce congestion, lower delay costs and enhance efficiency. Significant contribution has been made in the field of large scale optimization and integrated planning of operations in container terminals. Bulk ports on the other hand have received almost no attention in the operations research literature. The integrated berth allocation and quay crane assignment or scheduling problem has been studied in the past by Park and Kim (2003), Meisel and Bierwirth (2006), Imai, Chen, Nishimura and Papadimitriou (2008), Meisel and Bierwirth (2009), and more recently by Giallombardo et al. (2010) and Vacca (2011) for container terminals. Comprehensive literature surveys on container terminal operations can be found in Steenken et al. (2004), Stahlbock and Voss (2008), Bierwirth and Meisel (2010).

The dynamic, hybrid berth allocation problem in bulk ports is studied by Umang et al. (2013). A detailed literature survey on the existing studies related to the BAP in the context of container terminals is provided in Chapter 3.

Yard management in container terminals involves several tactical and operational level decision problems. Scheduling and deployment of yard cranes is addressed by Cheung et al. (2002), Zhang et al. (2002), Ng and Mak (2005), Ng (2005) and Jung and Kim (2006). Storage and space allocation, stacking and re-marshalling strategies have been studied by Kim and Kim (1999), Kim et al. (2003), Lee et al. (2006) and few others. Nishimura et al. (2009) investigate the storage plan for transshipment hubs, and propose an optimization model to minimize the sum of the waiting time of feeders and the handling times for transshipment containers flow. Transfer operations that consist of routing and scheduling of internal trucks, straddle carriers and AGV's have been studied by Liu et al. (2004), Vis et al. (2005), and Cheng et al. (2005) among others. Works on integrated problems related to yard management in container terminals in-

## Chapter 4. The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports

---

clude Bish et al. (2001) and Kozan and Preston (2006) who propose the integration of yard allocation and container transfers, whereas Chen et al. (2007) and Lau and Zhao (2007) study the integrated scheduling of handling equipment in a container terminal. In the following, we discuss in more detail some articles relevant to our study.

Moorthy and Teo (2006) discuss the concepts of berth template and yard template in the context of transshipment hubs in container shipping. They study the delicate trade-off between the level of service as indicated by the vessel waiting times and the operational cost for moving containers between the yard and quay in a container terminal. A robust berth allocation plan is developed using the sequence pair approach, with the objective to minimize the total expected delays and connectivity cost that is related to the distance between the berthing positions of vessels belonging to the same transshipment group.

Cordeau et al. (2007) study the Service Allocation Problem (SAP), a tactical problem arising in the yard management of Gioia Tauro Terminal. The SAP is a yard management problem that deals with dedicating specific areas of the yard and the quay to the services or route plans of shipping companies which are planned in order to match the demand for freight transportation. The objective of the SAP is the minimization of container rehandling operations in the yard and it is formulated as a Generalized Quadratic Assignment Problem (GQAP, see e.g. Cordeau et al. (2006), and Hahn et al. (2008)). An evolutionary heuristic is developed to solve larger instances obtained from the real port data.

Zhen, Chew and Lee (2011) propose a mixed integer model to simultaneously solve the tactical berth template and yard template planning in transshipment hubs. The objective is to minimize the sum of service cost derived from the violation of the vessels expected turnaround time intervals and the operation cost related to the route length of transshipment container flows in the yard. A heuristic algorithm is developed to solve large scale instances within reasonable time and numerical experiments are conducted on instances from real world data to validate the efficiency of the proposed algorithm.

More recently, Lee and Jin (2013) study the feeder vessel management problem in a container transshipment hub. The integrated problem consists of three tactical decision problems of berth template, schedule template and yard template design. The problem is formulated as a mixed integer program and solved using a memetic heuristic based on genetic algorithm and tabu search. The effectiveness of the proposed algorithm is validated by conducting numerical

experiments on instances based on real port data.

In the operations research literature, bulk terminals have thus far received almost no attention. In the context of container terminals, the major focus in the field of large scale optimization has been on studying the integrated berth allocation and quay crane scheduling or assignment problem, while very few scholars have attempted to solve the combined problem of berth allocation and yard assignment as a single large scale optimization problem. To the best of our knowledge, this is also the first study to present an exact solution algorithm (based on branch-and-price) to solve the integrated problem of berth allocation and yard assignment in the context of seaside port operations planning.

## 4.3 Problem Statement

In this section we elaborate on the background for the integrated berth allocation and yard assignment problem in the context of bulk ports. A schematic representation of a bulk port terminal is shown in Figure 4.1. We consider a set of vessels  $N$ , to be berthed on a continuous quay of length  $L$  over a time horizon  $H$ . We consider dynamic vessel arrivals and a hybrid berth layout in which the quay boundary is discretized into a set  $M$  of sections of variable lengths. In a feasible berthing assignment, a given vessel may occupy more than one section, however a given section cannot be occupied by more than one vessel or part of a vessel at any given time. The dynamic, hybrid berth allocation problem in bulk ports is studied by Umang et al. (2013), in which two alternative exact solution methods and a heuristic approach are proposed to solve the problem. The problem is solved for a given yard layout and the unit handling times for given sections along the quay and cargo types are provided as input parameters to the model. In the present work, we extend the berth allocation problem to account for the assignment of yard locations to specific cargo types and vessels berthing at the port which also become decision variables in the integrated framework. Evidently the integrated problem is much more complex and extensive than the berth allocation problem studied in Umang et al. (2013). The integrated problem is solved for a given time frame, and the objective is to minimize the service times of the vessels berthing at the port.

A major difference between bulk port and container terminal operations is the need to explicitly account for the cargo type on the vessel in bulk ports. Depending on the vessel requirements and cargo types, a wide variety of specialized equipment such as conveyors and pipelines are used for discharging

## Chapter 4. The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports

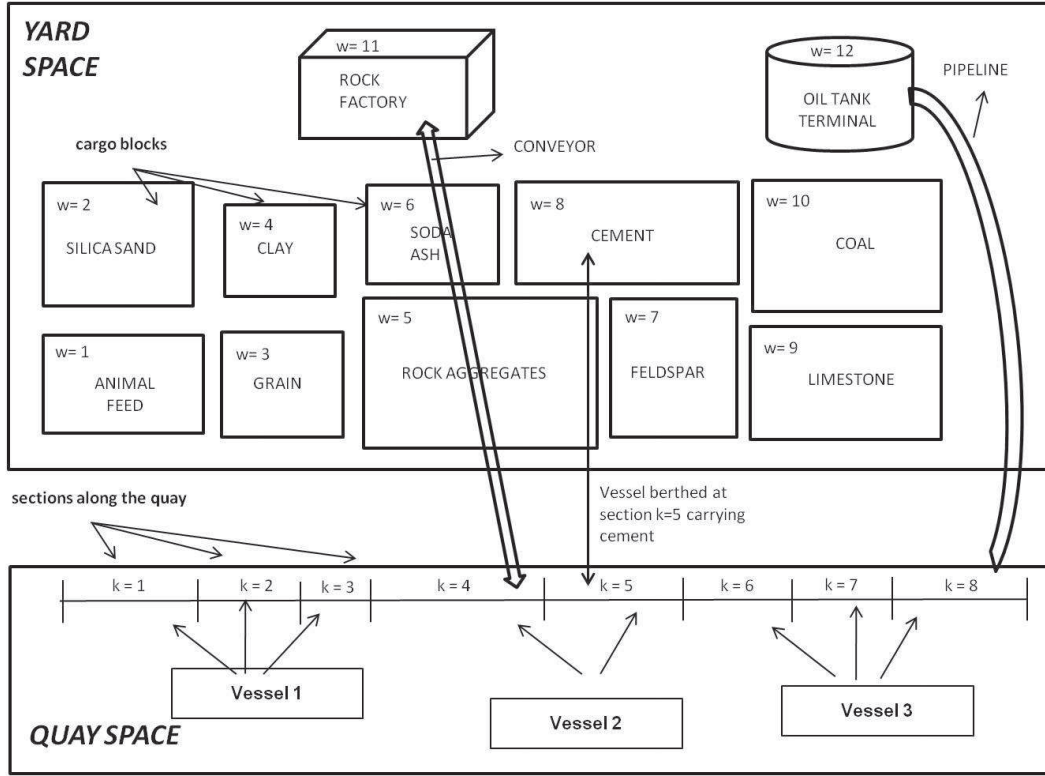


Figure 4.1: Schematic representation of a bulk port terminal

or loading operations. For example, liquid bulk is generally discharged using pipelines which are installed at only certain sections along the quay. Similarly, a vessel may require the conveyor facility to load cargo from a nearby factory outlet to the vessel. In contrast in a container terminal, all cargo is packed into containers, and thus there is no need for any specialized equipment to handle any particular type of cargo. Furthermore in bulk ports, depending on the cargo properties, there may be additional restrictions on the storage of specific cargo types in the yard which forbids two or more cargo types to be stored in adjacent yard locations to avoid intermixing.

In our model we assume a fixed crane deployment during the processing time of the vessel. Note that this assumption is consistent with our observations during our visit to the bulk port under study. It was observed that for certain cargo types, only conveyors and pipelines are used for the loading/ discharging operations and the cranes are not used at all. For the vessels which do require the cranes for handling operations, the port ensures a fixed number of cranes to operate on the vessel during the handling time of the vessel, where the number of cranes is a function of the length of the vessel. This is partly made possi-



ble by the fact that the total number of cranes available at the port is significantly higher than the number of cranes handling the vessels at any given time, thus enabling the port to replace cranes in the unlikely event of breakdown or other mechanical problems. Moreover the cranes can move freely and pass each other as opposed to the cranes in container terminals that are restricted to movements by rail. Thus in our model, we do not explicitly include constraints on the scheduling or assignment of cranes, since the number of cranes operating on a given vessel can be determined by data pre-processing depending on the cargo type and the length of the vessel.

It is further assumed that each vessel has a single cargo type that can be discharged (loaded) and transferred to (from) multiple yard locations. The assumption of single cargo type on each berthing vessel is consistent with the data sample that we received from the port. Moreover to model multiple cargo types on the same vessel, the internal structure of each vessel needs to be modeled more rigorously which is beyond the scope of this study, nor is typically considered in any studies related to berth allocation planning. The assumption that a given cargo type can be stored at several yard locations is true in practice. The cargo is stored at several yard locations to load (discharge) a single vessel, and moreover there are situations when more than one vessel with the same cargo type are being handled at different locations along the quay, in which case the same cargo type is stored at multiple locations on the yard as close to the berthed vessels as possible.

In the computation of the handling times, it is assumed that all sections occupied by the berthed vessel are being operated on simultaneously. The amount of cargo handled at each section is proportional to the length of the berthed section, and the handling time of the vessel is the time taken to load or discharge the section whose operation finishes last. We define the unit handling time of a vessel as the time taken to load (discharge) a unit quantity of cargo on (from) the vessel. The unit handling time of a vessel has a fixed component dependent on the number of quay cranes operating on the vessel, and a variable component which is dependent on the distance between the berthing location of the vessel and the storage location of the corresponding cargo type on the yard. Since a given cargo type can be discharged (loaded) and transferred to (from) multiple yard locations, the distance between the berthing location of the vessel and the storage location of the cargo type is the weighted average distance between the vessel and all the cargo locations that are assigned to the vessel. In this calculation, the weights are assumed equal to the cargo quantities that are transferred to (from) each yard location from (to) the vessel.

## Chapter 4. The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports

---

Another assumption in the model is that in the planning horizon, a given yard location is either assigned to a single cargo type, or alternatively the yard location is not assigned to any cargo type. This assumption is realistic because in periods of low congestion at the port, there may be several yard locations that are not assigned to any cargo type. Even in periods of congestion, as a standard practice at the port, two different cargo types are never stored at the same yard location. The assumption that the assignment of a cargo type to a specific storage location on the yard does not alter during the planning horizon makes sense because in our model, the two problems of berth allocation and yard assignment are solved simultaneously for a given time frame. This implies that the integrated model tries to determine the optimal assignment of cargo types to yard locations for which the handling times of the vessels berthing in that planning horizon are minimized. In practice when the model is implemented in a rolling planning horizon, there is a considerable gap of at least a few days between two planning runs. In that case the assignment of cargo types on the yard will change depending on the cargo types on the vessels berthing in the next planning window.

Based on the preceding discussion, the unit handling time  $h_{ik}^w$  for vessel  $i$  with cargo type  $w$  occupying section  $k$  along the quay includes the time taken to transfer the unit quantity of cargo between the cargo location on the yard and section  $k$ , and the time taken to load (or unload) the cargo from the quay side to the vessel. These can be denoted by  $\beta_{ik}^w$  and  $\alpha_{ik}^w$  respectively. Thus we have,  $h_{ik}^w = \alpha_{ik}^w + \beta_{ik}^w$ , where  $\alpha_{ik}^w = T/n_{ik}^w$  and  $\beta_{ik}^w = V_w e_k^i$ . Here  $T$  is the amount of time taken by a single crane to load or discharge a unit quantity of cargo, and  $n_{ik}^w$  is the number of cranes operating in section  $k$  on vessel  $i$  for cargo type  $w$ .  $\beta_{ik}^w$  is the time taken to transfer a unit quantity of cargo between the cargo location  $w$  on the yard and the section  $k$  for vessel  $i$ , which is assumed to be a linear function of the weighted average distance  $e_k^i$  between the section  $k$  and all cargo locations assigned to the vessel  $i$ . The parameter  $V_w$  depends on the rate of transfer of cargo type  $w$ . Thus for example, if a vessel is using the conveyor facility to load rock aggregates from the rock factory directly into the vessel, the parameter  $V_w$  is equal to the cargo transfer rate for the conveyor facility, and if there are no additional cranes operating on the vessel, the parameter  $\alpha_{ik}^w$  which is provided as an input parameter to the model is equal to zero. In practice, the fixed specialized equipment facilities such as conveyors and pipelines are dedicated to handling certain cargo types. For example, liquid bulk is transferred using pipelines, and rock aggregates are transferred using conveyor facilities. Thus the specialized facilities are themselves modeled as cargo types in the proposed model. The objective of the integrated optimization model that we solve is to

minimize the sum of the service times of all vessels, which includes the handling or processing times and the berthing delays for all vessels berthing at the port.

## 4.4 Model Formulation

In this section, we present a mixed integer programming formulation for the integrated berth allocation and yard assignment problem in bulk ports.

### 4.4.1 Notation

**Input parameters** In the formulation of the integrated model, the following input data is assumed available:-

$N =$	set of vessels
$M =$	set of sections
$P =$	set of cargo locations
$W =$	set of cargo types
$H =$	set of time steps
$W_i =$	cargo type to be loaded or discharged from vessel $i$
$\bar{P}(p) =$	set of cargo locations neighbouring cargo location $p$
$\bar{W}(w) =$	set of cargo types that cannot be stored adjacent to cargo type $w$
$A_i =$	expected arrival time of vessel $i$
$D_i =$	draft of vessel $i$
$L_i =$	length of vessel $i$
$Q_i =$	quantity of cargo for vessel $i$
$d_k =$	draft of section $k$
$\ell_k =$	length of section $k$
$b_k =$	starting coordinate of section $k$
$\alpha_{ik}^w =$	deterministic component of handling time for cargo type $w$ of vessel $i$ berthed at section $k$
$V_w =$	constant dependent on the rate of transfer of cargo type $w$
$r_k^p =$	distance between cargo location $p$ and section $k$
$R_w =$	maximum amount of cargo type $w$ that can be handled in a single time step
$L =$	total length of quay
$B =$	large positive constant
$F =$	maximum number of cargo locations that can be assigned to a single vessel

## Chapter 4. The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports

---

$$\rho_{i\ell k} = \text{fraction of cargo handled at section } k \text{ when section } \ell \text{ is the first section occupied by vessel } i$$

$$\delta_{i\ell k} = \begin{cases} 1 & \text{if vessel } i \text{ starting at section } \ell \text{ touches section } k \\ 0 & \text{otherwise.} \end{cases}$$

**Decision Variables** The following decision variables are used in the model:-

- $m_i$  the starting time of handling of vessel  $i \in N$ ;
- $c_i$  the total handling time of vessel  $i \in N$ ;
- $h_{ik}^w$  handling time for unit quantity of cargo type  $w$  for vessel  $i$  berthed at section  $k$  ;
- $\beta_{ik}^w$  variable component of handling time of vessel  $i$  with cargo type  $w$  berthed at section  $k$  along the quay;
- $q_{ip}$  amount of cargo handled by vessel  $i$  at cargo location  $p$ ;
- $\eta_i$  number of cargo locations assigned to vessel  $i$ ;
- $e_k^i$  weighted average distance between section  $k$  occupied by vessel  $i$  and all cargo locations assigned to the vessel;
- $s_k^i$  binary, equals 1 if section  $k \in M$  is the starting section of vessel  $i \in N$ , 0 otherwise;
- $x_{ik}$  binary, equals 1 if vessel  $i \in N$  occupies section  $k \in M$ , 0 otherwise;
- $y_{ij}$  binary, equals 1 if vessel  $i \in N$  is berthed to the left of vessel  $j \in M$  without any overlapping in space, 0 otherwise;
- $z_{ij}$  binary, equals 1 if handling of vessel  $i \in N$  finishes before the start of handling of vessel  $j \in N$ , 0 otherwise;
- $\mu_w^p$  binary, equals 1 if cargo type  $w$  is stored at cargo location  $p$ , 0 otherwise;
- $\phi_{ip}$  binary, equals 1 if cargo location  $p$  is assigned to vessel  $i$ , 0 otherwise;
- $\theta_{it}$  binary, equals 1 if vessel  $i$  is being handled at time  $t$ , 0 otherwise;
- $\omega_t^{ip}$  binary, equals 1 if vessel  $i$  is being handled at location  $p$  at time  $t$ , 0 otherwise;

### 4.4.2 Mathematical Model

The integrated model for the berth allocation and yard assignment problem in bulk ports can be formulated as follows:

$$\min \sum_{i \in N} (m_i - A_i + c_i) \quad (4.1)$$

$$\text{s.t. } m_i - A_i \geq 0 \quad \forall i \in N \quad (4.2)$$

$$\sum_{k \in M} (s_k^j b_k) + B(1 - y_{ij}) \geq \sum_{k \in M} (s_k^i b_k) + L_i \quad \forall i, j \in N, i \neq j \quad (4.3)$$

$$m_j + B(1 - z_{ij}) \geq m_i + c_i \quad \forall i \in N, \forall j \in N, i \neq j \quad (4.4)$$

$$y_{ij} + y_{ji} + z_{ij} + z_{ji} \geq 1 \quad \forall i \in N, \forall j \in N, i \neq j \quad (4.5)$$

$$\sum_{k \in M} s_k^i = 1 \quad \forall i \in N \quad (4.6)$$

$$\sum_{k \in M} (s_k^i b_k) + L_i \leq L \quad \forall i \in N \quad (4.7)$$

$$\sum_{\ell \in M} (\delta_{i\ell k} s_\ell^i) = x_{ik} \quad \forall i \in N, \forall k \in M \quad (4.8)$$

$$(d_k - D_i) x_{ik} \geq 0 \quad \forall i \in N, \forall k \in M \quad (4.9)$$

$$c_i \geq h_{ik}^w \rho_{ik} Q_i - B(1 - s_\ell^i) \quad \forall i \in N, \forall \ell \in M, \forall k \in M, \forall w \in W \quad (4.10)$$

$$h_{ik}^w = \alpha_{ik}^w + \beta_{ik}^w \quad \forall w \in W, \forall i, \forall k \in M \quad (4.11)$$

$$\beta_{ik}^w = V_w e_k^i \quad \forall i \in N, \forall w \in W, \forall k \in M \quad (4.12)$$

$$e_k^i = \sum_{p \in P} (r_k^p q_{ip}) / Q_i \quad \forall i \in N, \forall k \in M \quad (4.13)$$

$$Q_i = \sum_{p \in P} q_{ip} \quad \forall i \in N \quad (4.14)$$

$$q_{ip} \leq \phi_{ip} Q_i \quad \forall i \in N, \forall p \in P \quad (4.15)$$

$$\phi_{ip} \leq q_{ip} \quad \forall i \in N, \forall p \in P \quad (4.16)$$

$$q_{ip} \leq \sum_{w \in W_i} \sum_{t \in H} (R_w \omega_t^{ip} + B(1 - \mu_w^p)) \quad \forall i \in N, \forall p \in P \quad (4.17)$$

$$\sum_{p \in P} \phi_{ip} \leq F \quad \forall i \in N \quad (4.18)$$

$$\mu_w^p + \mu_{\bar{w}}^{\bar{p}} \leq 1 \quad \forall w \in W, \forall \bar{w} \in \bar{W}(w), \forall p \in P, \forall \bar{p} \in \bar{P}(p) \quad (4.19)$$

$$\sum_{i \in N} \omega_t^{ip} \leq 1 \quad \forall p \in P, \forall t \in H \quad (4.20)$$

$$\sum_{w \in W} \mu_w^p \leq 1 \quad \forall p \in P \quad (4.21)$$

$$\phi_{ip} \leq \mu_w^p \quad \forall i \in N, \forall w \in W_i, \forall p \in P \quad (4.22)$$

$$\omega_t^{ip} \geq \phi_{ip} + \theta_{it} - 1 \quad \forall i \in N, \forall p \in P, \forall t \in H \quad (4.23)$$

$$\omega_t^{ip} \leq \phi_{ip} \quad \forall i \in N, \forall p \in P, \forall t \in H \quad (4.24)$$

## Chapter 4. The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports

$$\omega_t^{ip} \leq \theta_{it} \quad \forall i \in N, \forall p \in P, \forall t \in H \quad (4.25)$$

$$\sum_{t \in H} \theta_{it} = c_i \quad \forall i \in N \quad (4.26)$$

$$t + B(1 - \theta_{it}) \geq m_i + 1 \quad \forall i \in N, \forall t \in H \quad (4.27)$$

$$t \leq m_i + c_i + B(1 - \theta_{it}) \quad \forall i \in N, \forall t \in H \quad (4.28)$$

$$s_k^i, x_{ik} \in \{0, 1\} \quad \forall i \in N, \forall k \in M \quad (4.29)$$

$$y_{ij}, z_{ij} \in \{0, 1\} \quad \forall i, j \in N \quad (4.30)$$

$$\mu_w^p \in \{0, 1\} \quad \forall p \in P, \forall w \in W \quad (4.31)$$

$$\omega_t^{ip} \in \{0, 1\} \quad \forall i \in N, \forall p \in P, \forall t \in H \quad (4.32)$$

$$\phi_{ip} \in \{0, 1\} \quad \forall i \in N, \forall p \in P \quad (4.33)$$

$$\theta_{it} \in \{0, 1\} \quad \forall i \in N, \forall t \in H \quad (4.34)$$

The objective function (4.1) minimizes the total service time of all vessels, which is the sum of total delays and total handling time of vessels berthing at the port. Constraints (4.2) are the dynamic arrival constraints that ensure that vessels can be serviced only after their arrival. Constraints (4.3)-(4.5) are the non-overlapping restrictions for any two vessels berthing at the port. These ensure that while two vessels may be overlapping in space or in time, they cannot be simultaneously overlapping in both space and time. Note that the constraints (4.3)-(4.4) have been linearized by using a large positive constant  $B$ . Constraints (4.6)-(4.8) ensure that each vessel occupies only as many number of sections as determined by its length and the starting section occupied by the vessel. Constraints (4.9) ensure that the draft of the vessel does not exceed the draft of any occupied section.

Constraints (4.10) are used to determine the total handling time for any given vessel which is equal to the time taken to process the section occupied by the vessel whose operation finishes last. The variable component of the handling time for a given vessel  $i$  and given occupied section  $k$  is determined by the constraint (4.12), and is a function of the weighted average distance between section  $k$  and all cargo locations assigned to vessel  $i$ . The average distance is weighted over the cargo quantities transferred between each cargo location  $p$  assigned to vessel  $i$  and section  $k$  occupied by the vessel, and is determined by constraint (4.13). Constraints (4.11) determine the unit handling time of vessel  $i$  at a given section  $k$ , which is the sum of the fixed component dependent on the number of cranes operating on that section and the variable component of the handling time as discussed earlier. Constraints (4.14)-(4.16) state that the

total cargo quantity to be loaded (discharged) is equal to the sum of the cargo quantities transferred from (to) all the cargo locations assigned to the vessel. Constraints (4.17) are capacity constraints to ensure that the amount of cargo transferred in a unit time does not exceed the maximum amount of cargo that can be handled as given by parameter  $R_w$  for cargo type  $w$ . Note that in case of specialized equipment facilities, the parameter  $R_w$  may refer to the conveyor speed or the flow rate through the pipeline; in other cases it may refer to the maximum rate of transfer of material between the quay side and the yard side that can be achieved by assigning auxiliary equipment such as loading shovels, wheel loaders etc.

Constraints (4.18) impose an upper bound on the maximum number of cargo locations that can be assigned to any single vessel. Constraints (4.19) ensure that two cargo types that cannot be stored together, for example coal and clay, are not assigned to adjacent yard locations. Constraints (4.20) state that a given cargo location at a given time can be used by at most one vessel to avoid a congestion. Constraints (4.21) state that the yard locations have dedicated cargo types, and a given yard location can be assigned to at most one cargo type. Constraints (4.22) ensure that a vessel is assigned to a yard location only if that yard location stores the cargo type on the vessel.

Constraints (4.23) - (4.25) control the values of the binary decision variable  $\omega_t^{ip}$  which should take value equal to 1, if and only if both the binary variables  $\phi_{ip}$  and  $\theta_{it}$  are equal to 1. This implies that vessel  $i$  is handled at cargo location  $p$  at time  $t$  if and only if, cargo location  $p$  is assigned to vessel  $i$  and vessel  $i$  is being handled at time  $t$ . Similarly, constraints (4.26) - (4.28) control the values of the binary decision variable  $\theta_{it}$  which is equal to 1 at all time intervals between the starting time of the handling of the vessel and the finishing time of the handling operations, and 0 otherwise.

## 4.5 Exact Solution Approach

The mixed integer programming formulation of the integrated model is extremely complex and unwieldy, and could not be used to solve even small sized instances, as later validated by numerical experiments. In the following section, we decompose the mixed integer model and formulate it as a set partitioning problem.

### 4.5.1 Set Partitioning Model

Let  $\Omega$  be the set of feasible berthing assignments of all vessels berthing at the port in the given planning horizon. Note that a feasible assignment represents the assignment of a single vessel for a given set of section(s) for a specific time period, assigned to a set of specific cargo location(s) in the yard. Let  $c_a$  be the cost of assignment  $a \in \Omega$ . The following input parameters are used in the set partitioning model:

#### Input Parameters

$$A_a^i = \begin{cases} 1 & \text{if vessel } i \text{ is assigned in assignment } a, \\ 0 & \text{otherwise.} \end{cases} \quad (4.35)$$

$$B_a^{kt} = \begin{cases} 1 & \text{if section } k \text{ is occupied at time } t \text{ in assignment } a, \\ 0 & \text{otherwise.} \end{cases} \quad (4.36)$$

$$C_a^{pw} = \begin{cases} 1 & \text{if cargo type } w \text{ is stored at location } p \text{ in assignment } a, \\ 0 & \text{otherwise.} \end{cases} \quad (4.37)$$

$$D_a^{pt} = \begin{cases} 1 & \text{if assignment } p \text{ is handled at cargo location } p \text{ at time } t, \\ 0 & \text{otherwise.} \end{cases} \quad (4.38)$$

There is one additional parameter  $\vartheta_w$ , which indicates the number of vessels carrying cargo type  $w \in W$ . The set partitioning model for the integrated berth allocation and yard assignment problem can then be formulated as follows:

$$\min \sum_{a \in \Omega} c_a \lambda_a \quad (4.39)$$

$$\text{s.t. } \sum_{a \in \Omega} A_a^i \lambda_a = 1, \quad \forall i \in N, \quad (4.40)$$

$$\sum_{a \in \Omega} B_a^{kt} \lambda_a \leq 1, \quad \forall k \in M, \forall t \in H, \quad (4.41)$$

$$\sum_{a \in \Omega} C_a^{pw} \lambda_a - \vartheta_w \mu_w^p \leq 0, \quad \forall p \in P, \forall w \in W, \quad (4.42)$$

$$\sum_{w \in W} \mu_w^p \leq 1, \quad \forall p \in P, \quad (4.43)$$



$$\mu_w^p + \mu_{\bar{w}}^{\bar{p}} \leq 1, \quad \forall p \in P, \forall \bar{p} \in \bar{P}(p), \forall w \in W, \forall \bar{w} \in \bar{W}(w), \quad (4.44)$$

$$\sum_{a \in \Omega} D_a^{pt} \lambda_a \leq 1, \quad \forall p \in P, \forall t \in H, \quad (4.45)$$

$$\lambda_a \in \{0, 1\}, \quad \forall a \in \Omega, \quad (4.46)$$

$$\mu_w^p \in \{0, 1\}, \quad \forall p \in P, \forall w \in W. \quad (4.47)$$

In the above formulation,  $\lambda_a$  indicates if assignment  $a \in \Omega$  is part of the optimal solution (i.e.  $\lambda_a = 1$ ), and the decision variable  $\mu_w^p$  is retained from the original formulation and indicates if location  $p \in P$  stores cargo type  $w \in W$ .

The objective function (4.39) minimizes the total service time of vessels berthing at the port. Constraints (4.40) ensure that there is exactly one feasible berthing assignment for each vessel in the optimal solution. Constraints (4.41) state that a given section at a given time can be occupied by at most one vessel. Constraints (4.42) and (4.43) state that a given cargo location on the yard can store at most one cargo type. Constraints (4.44) are identical to constraints (4.19) in the original formulation, ensuring that cargo types that cannot be stored together are not stored at adjacent locations in the yard. Constraints (4.45) ensure that at most one vessel can be handled at a cargo location at a given time to avoid congestion. Constraints (4.46) and (4.47) state that both the decision variables  $\lambda_p$  and  $\mu_w^p$  are binary integer variables and can only take (0,1) values.

The above formulation was tested for small instances inspired from real port data. However, the number of feasible assignments given by the size of  $(\Omega)$  grows exponentially with the problem size. This in turn leads to an exponential growth in the computational time. Hence in order to avoid the "explosion" of the solution time, we propose to solve the linear programming relaxation of the above problem using column generation, as described in the next section.

### 4.5.2 Column Generation

In the linear programming (LP) relaxation of the set partitioning problem the domains of  $\lambda_a$  and  $\mu_w^p$  are extended to  $[0, 1]$ . Despite the large number of variables it is possible to solve LP relaxation using a column generation algorithm. In a column generation algorithm we maintain a restricted master problem (RMP) that only considers a subset  $\Omega_1 \subseteq \Omega$  of all the possible variables. New variables are added to  $\Omega_1$  until no variable in  $\Omega \setminus \Omega_1$  can further improve the

## Chapter 4. The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports

---

### Algorithm 3: Branch and Price

---

**Data:** data file,  $\Omega$ , finished - boolean, duals - float

**Result:**  $\Omega_1 \subset \Omega$ , solution

```
1 begin
2    $\Omega_1 \leftarrow \text{initialHeuristic}(\Omega)$ 
3   duals  $\leftarrow \emptyset$ 
4   solution  $\leftarrow \emptyset$ 
5   repeat
6     duals  $\leftarrow \text{solveMaster}(\Omega_1)$ 
7     finished  $\leftarrow \text{true}$ 
8     for  $i \in N$  do
9       temp  $\leftarrow \text{solveSubProblem}(i, \text{duals})$ 
10      if  $\text{reducedCost}(\text{temp}) < 0$  then
11         $\Omega_1 = \Omega_1 \cup \text{temp}$ 
12        finished  $\leftarrow \text{false}$ 
13  until finished
14  solution  $\leftarrow \text{solveMaster}(\Omega_1)$ 
15  if solution  $\notin \mathbb{Z}$  then
16    ub  $\leftarrow \text{solveMaster}(\Omega_1, \text{integral})$ 
17    if solution = ub then
18      break
19    solution  $\leftarrow \text{branch\&bound}(\text{solution})$ 
20  print solution
```

---

solution that results from only using the variables in  $\Omega_1$ .

In the first iteration of column generation, the RMP is solved using the set  $\Omega_1$  consisting of vessel assignments in the initial feasible solution provided by the Initial Solution Heuristic (Section 20). Thereafter, in each successive iteration of the column generation process, the following dual variables are passed to the sub-problem for identifying feasible assignments with negative reduced cost:

- $\pi_i$  - dual variables corresponding to constraints (4.40)
- $\tau_{kt}$  - dual variables corresponding to constraints (4.41)
- $\xi_{pw}$  - dual variables corresponding to constraints (4.42)
- $\gamma_{pt}$  - dual variables corresponding to constraints (4.45)

We do not need to consider the dual variables corresponding to constraints (4.43)-(4.44) since these constraints do not involve the  $\lambda_a$  variables and therefore the associated dual variables do not impact the reduced cost of the  $\lambda_a$  variables. Based on the dual variables from RMP, the sub-problem generates new columns to enter the active pool of columns  $\Omega_1$  by calculating the most negative reduced cost column for each vessel separately in each iteration of the column generation process. When there are no columns with negative reduced cost for any sub-problem to enter  $\Omega_1$ , the column generation terminates.

The column generation in pseudo-code can be seen in Lines (1)-(13) in Algorithm 3. For mathematical justification of column generation, please refer to Barnhart et al. (1998), Desaulniers et al. (2005) and Feillet (2010).

### Initial Solution

In order to execute the column generation, an initial feasible solution is needed, or alternatively artificial high cost variables can be added to the master problem. Note that if the problem is infeasible, no dual variables are produced and column generation fails. An *Initial Solution Heuristic Algorithm* shown in Algorithm 4 is designed to extract an initial feasible solution to the master problem. The heuristic searches through the space of all feasible columns ( $C$ ), where a column represents the assignment of a single vessel and includes information about the set of sections occupied by the vessel, the set of cargo locations in

## Chapter 4. The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports

---

### Algorithm 4: Initial Solution Heuristic

---

**Data:**  $N$  - set of vessels,  $C$  - set of columns

**Result:**  $\pi$  – initial solution

```

1 begin
2    $\pi \leftarrow \emptyset$ 
3   for  $i \in N$  do
4     for  $j \in C$  do
5       if  $\pi = \emptyset$  then
6          $\pi \leftarrow j$ 
7         break
8       else if  $i \in j$  and  $j$  is compatible with  $\pi$  then
9          $\pi = \pi \cup j$ 
10        break
11  return  $\pi$ 

```

---

the yard assigned to the vessel and the time period for which the given set of section(s) and the cargo location(s) in the yard are occupied by the vessel.

### Sub-Problem

In each iteration of column generation, we solve  $|N|$  sub-problems, one for each vessel  $i \in N$ . In each sub-problem, the objective is to identify the feasible assignment for that particular vessel with the most negative reduced cost to be added to the current pool of active columns  $\Omega_1$  in the restricted master problem. Note that the index  $i \in N$  is removed from all decision variables and input parameters in the sub-problem, since it is solved separately for each vessel  $i \in N$ . The objective function in a single sub-problem can be written as:

$$\min (m - a + c) - (\pi + \sum_{k \in M} \sum_{t \in H} \tau_{kt} \cdot \sigma_{kt} + \sum_{p \in P} \sum_{t \in H} \gamma_{pt} \cdot \psi_{pt} + \sum_{p \in P} \sum_{w \in W} \xi_{pw} \cdot \mu_{pw}) \quad (4.48)$$

where:

**Input Parameters** The input parameters used in the sub-problem are:

$\pi, \tau_{kt}, \gamma_{pt}, \xi_{pw}$  the dual variables obtained from the restricted master problem

$a$	the arrival time of the vessel
$w$	cargo type on the vessel
$\delta_{lk}$	fraction of cargo handled at section $k$ , if the starting section of the vessel is $l$
$B$	large positive constant (set to 1,000,000)
$b_k$	starting coordinate of section $k$
$L$	length of the vessel
$Q$	quantity of cargo on the vessel
$Y$	quay length
$\rho_{lk}$	binary, equals 1 if section $k$ is occupied by the vessel when section $l$ is the starting section, 0 otherwise
$F$	maximum number of cargo locations that can be assigned to the vessel
$r_k^p$	distance between section $k$ and cargo location $p$
$\alpha_k^w$	number of cranes operating in section $k$ for cargo type $w$
$V_w$	transfer rate of cargo type on the vessel

**Decision Variables** The decision variables used in the sub-problem are:

$m$	integer $\geq 0$ , the starting time of handling the vessel
$c$	integer $\geq 0$ , the handling time of the vessel
$h_k^w$	handling time of the vessel at section $k$
$q_p$	quantity of cargo handled by the vessel at cargo location $p$
$r_k$	weighted average distance for section $k$ occupied by the vessel
$s_l$	binary, equals 1 if section $l$ is the starting section of the vessel, 0 otherwise
$x_k$	binary, equals 1 if section $k$ is occupied by the vessel, 0 otherwise
$\phi_p$	binary, equals 1 if cargo location $p$ is assigned to the vessel, 0 otherwise
$\theta_t$	binary, equals 1 if the vessel is served at time $t$ , 0 otherwise
$\sigma_{kt}$	binary, equals 1 if section $k$ is occupied at time $t$ , 0 otherwise
$\psi_{pt}$	binary, equals 1 if cargo location $p$ is used at time $t$ , 0 otherwise
$\mu_w^p$	binary, equals 1 if 1 if cargo type $w$ is stored at location $p$ , 0 otherwise

**Sub-problem formulation** The subproblem can be formulated as a mixed integer linear program as follows:

## Chapter 4. The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports

---

$$\min (m - a + c) - (\pi + \sum_{k \in M} \sum_{t \in H} \tau_{kt} \cdot \sigma_{kt} + \sum_{p \in P} \sum_{t \in H} \gamma_{pt} \cdot \psi_{pt} + \sum_{p \in P} \sum_{w \in W} \xi_{pw} \cdot \mu_w^p) \quad (4.49)$$

$$m - a \geq 0, \quad (4.50)$$

$$c \geq h_k^w \rho_{lk} - B(1 - s_l), \quad \forall l, k \in M \quad (4.51)$$

$$\sum_{l \in M} s_l = 1, \quad (4.52)$$

$$\sum_{k \in M} (s_k b_k) + L \leq Y, \quad (4.53)$$

$$\sum_{l \in M} \rho_{lk} s_l = x_k, \quad \forall k \in M, \quad (4.54)$$

$$\sum_{l \in M} \phi_l \leq F, \quad (4.55)$$

$$\phi_p \leq \mu_w^p, \quad \forall p \in P, \quad (4.56)$$

$$\sum_{p \in P} q_p = Q, \quad (4.57)$$

$$q_p \leq \phi_p Q, \quad \forall p \in P, \quad (4.58)$$

$$\phi_p \leq q_p, \quad \forall p \in P, \quad (4.59)$$

$$r_k = \left( \sum_{p \in P} r_k^p q_p \right) / Q, \quad \forall k \in M, \quad (4.60)$$

$$h_k^w = \alpha_k^w + V_w r_k, \quad \forall k \in M, \quad (4.61)$$

$$\sum_{t \in T} \theta_t = c, \quad (4.62)$$

$$t + B(1 - \theta_t) \geq m + 1, \quad \forall t \in H, \quad (4.63)$$

$$t \leq m + c + B(1 - \theta_t), \quad \forall t \in H, \quad (4.64)$$

$$\sigma_{kt} \geq x_k + \theta_t - 1, \quad \forall k \in M, \forall t \in H, \quad (4.65)$$

$$\sigma_{kt} \leq x_k, \quad \forall k \in M, \forall t \in H, \quad (4.66)$$

$$\sigma_{kt} \leq \theta_t, \quad \forall k \in M, \forall t \in H, \quad (4.67)$$

$$\psi_{pt} \geq \phi_p + \theta_t - 1, \quad \forall p \in P, \forall t \in H, \quad (4.68)$$

$$\psi_{pt} \leq \phi_p, \quad \forall p \in P, \forall t \in H, \quad (4.69)$$

$$\psi_{pt} \leq \theta_t, \quad \forall p \in P, \forall t \in H. \quad (4.70)$$

Constraints (4.50) ensure that vessels can be served after their arrival. Constraints (4.51) are used to determine the total handling time of the vessel. Constraints (4.52) state that each vessel has exactly one starting section. Constraints (4.53) state that the vessel should be berthed such that it does not extend beyond the length of the quay. Constraints (4.54) determine if a particular section is occupied by the vessel. Constraints (4.55) impose an upper bound on the number of cargo locations that can be assigned to a single vessel. Constraints (4.56) ensure that a vessel is assigned to a yard location only if that yard location stores the cargo type on the vessel. Constraints (4.57)-(4.59) state that the total cargo quantity to be loaded (discharged) is equal to the sum of the cargo quantities transferred from (to) all the cargo locations assigned to the vessel. Constraints (4.60) calculate the weighted average distance over cargo quantities and constraints (4.61) calculate the handling time for a given vessel and berthed section. Constraints (4.62)-(4.64) control the values of the binary decision variable  $\theta_t$  ensuring they take value equal to 1 at all times when the vessel is berthed along the quay. Constraints (4.63)-(4.65) control the values of the binary decision variable  $\sigma_{kt}$  to ensure that they take value equal to 1 if and only if both  $x_k$  and  $\theta_t$  are equal to 1. Similarly, constraints (4.66)-(4.68) control the values of the binary decision variable  $\psi_{pt}$  to ensure that they take value equal to 1 if and only if both  $\phi_p$  and  $\theta_t$  are equal to 1.

Since the sub-problem is solved separately for each vessel, the complexity of the problem is significantly reduced compared to the original problem. Thus in the proposed algorithm, the sub-problems are solved using an MIP solver.

### 4.5.3 Branch and Bound

Since we solve the linear relaxation of the restricted master problem, the final solution obtained after the convergence of the column generation process is typically not an integer solution. In the "worst" case, the solution of the relaxed version of the restricted master problem provides a lower bound to the solution of the original problem and an integer solution can be obtained by applying the branch and bound algorithm (for more details on branch and bound, please refer to Hillier and Lieberman (2001)) to the obtained solution. However, this does not guarantee that we obtain the optimal integer solution to the original problem, since there might be a column that would price out favorably but is

## Chapter 4. The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports

---

not present in the final pool of active columns  $\Omega_1$ . Therefore to find the optimal solution, column generation has to be executed again at every node of the branch and bound tree (Barnhart et al. (1998)).

The branching is held on the 2 decision variables:  $\lambda$  and  $\mu$ . Since  $\lambda$  variables are not restricting the solution space dramatically, we branch first on the  $\mu$  variables. Since  $\mu$  variables restrict the cargo type stored in a given cargo location, several columns are discarded, hence the convergence of branch and bound is faster than with the restriction of  $\lambda$  variables. When the list of  $\mu$  variables has been exhausted, i.e. there are no fractional  $\mu$  variables, then we start branching on the  $\lambda$  variables.

Note that while branching, both the RMP and the sub-problem(s) have to be modified by adding extra constraints. In the RMP, when  $\mu$  variables are set to 0 or 1 in a particular iteration, constraints enforcing the  $\mu$  variables to take the same values are also added in each sub-problem in that iteration. When  $\lambda_a$  is set to zero, it is ensured that the sub-problem for the vessel  $i$  in assignment  $a$  does not generate that assignment. Alternatively when  $\lambda_a$  is set to 1, the sub-problem for the vessel  $i$  in assignment  $a$  is not solved.

The combination of column generation and branch and bound is called branch-and-price. The complete algorithm for branch-and-price can be seen in Algorithm 3.

### 4.6 Improvement Methods

The branch-and-price method has been designed to tackle computationally heavy problems and to achieve a tight bound (lower bound in case of minimization). The convergence of the method is very much dependent on how well the decomposition of the original problem is carried out. The recommended method is to decompose the model in a way such that the sub-problem has a block angular structure (see Alvelos (2005) and Desaulniers et al. (2005)), so that each sub-problem can be solved independently of each other. In this study, each sub-problem is trying to find the berthing and yard assignment of a given vessel that has the most negative reduced cost for a given set of dual variables reported from the solution output of the restricted master problem in the last iteration. It is straightforward to see that the berthing schedule of one vessel (output of the specific sub-problem) solved separately, does not have any effect on the remaining sub-problems. Since the effect of dependency rises only in the



master problem, block angular structure is achieved for our problem.

Since we have access to  $\Omega$  (from the generation of the initial solution), the first strategy considered was to solve the master problem on  $\Omega_1$ , retrieve dual variables and calculate reduced costs on  $\Omega$ . However this method proved to be computationally heavy.

Another acceleration strategy to speed up the rate of convergence of the column generation process is to retrieve more than one negative reduced cost columns per sub-problem. This is achieved via modification of the original formulation of the sub-problem(s). The objective function is removed, implying that the problem has no objective and we are only searching for feasible solution(s)), and the following constraint is instead inserted into the model:

$$(\mathbf{m} - \mathbf{a} + \mathbf{c}) - (\pi + \sum_{k \in K} \sum_{t \in H} \tau_{kt} \cdot \sigma_{kt} + \sum_{p \in P} \sum_{t \in H} \gamma_{pt} \cdot \psi_{pt} + \sum_{p \in P} \sum_{w \in W} \xi_{pw} \cdot \mu_{pw}) \leq \epsilon \quad (4.71)$$

The above constraint together with constraints (4.50)-(4.70) yield as solution any feasible assignment with reduced cost less than zero ( $\epsilon$  being a very small number, e.g. 0.00001). We then use CPLEX functions to access the solution pool, containing all the feasible columns. Moreover in our case, we restrict the size of this pool to the maximum of 40 columns.

There are many other techniques to improve the convergence of the column generation process such as stabilization of the dual variables (Pigatti et al. (2005)) and dynamic constraint aggregation (Elhallaoui et al. (2005) and Elhallaoui et al. (2008)).

These methods are not universal and depend on the structure of the specific problem under study. We studied these two techniques for our problem, however the results didn't look very promising and hence they have not been included. In the dual stabilization method, the basic assumption that the dual variables are oscillating is not fulfilled and hence it leads to the failure of the method, although it needs to be a smaller number of columns to finish the column generation process.

The dynamic constraint aggregation technique that we tested is based on the aggregation of constraints of the same class and same coefficient values in the restricted master problem. This technique didn't speed up the convergence of the

column generation process significantly (except in the case of large instances), though the upper bounds were found to be tighter. This can be attributed to the fact that most of the computation time needed for the convergence of the column generation process is the time taken to solve the sub-problems, while the computation time of the master problem is already very fast for smaller problem size.

### 4.7 Meta-heuristic Approach

To provide an alternative to the exact solution algorithm proposed earlier, and obtain sub-optimal solutions in small computation time, we now propose a meta-heuristic to solve the integrated problem of berth allocation and yard assignment. In the existing literature, the proposed meta-heuristic is called critical-shaking neighborhood search (CSNS). The method was initially proposed by Lim and Xu (2006) to solve the yard allocation problem. The CSNS is based on deriving a feasible solution from a sequence of vessels, and iteratively changing the sequence of vessels to improve the solution quality. However unlike the squeaky wheel optimization heuristic described in Chapter 3 which is based on increasing priorities for the critical requests in the overall solution, the CSNS randomly shakes the top few critical requests to obtain a new priority list in order to escape local optima. In Lim and Xu (2006), the authors also apply a local search to the new priority list to find better candidate solutions.

An initial feasible solution is obtained using the *Initial Solution Heuristic* described in Algorithm 4. Based on the solution, a priority order for assigning vessels is obtained by assessing the service quality of each vessel, such that the worst performing vessel is ranked highest in the priority list followed by the second worst performing vessel and so on. The service quality of a vessel is based on the service time of the vessel in the solution. To prevent the algorithm from getting stuck at local minima, the priority rankings of the top few critical vessels (30 % in our implementation) are changed by swapping two randomly selected vessels. The new priority list obtained is then used to obtain a new solution using a *construction heuristic* that returns a feasible solution for a given priority order of vessels, as described in Algorithm 6. In the algorithm shown,  $\Omega$  refers to the set of all feasible assignments and  $p$  denotes the priority sequence. In the construction process, the *construction heuristic* tries to find the best feasible assignment for the given vessel  $i$  in the sequence from the set of all feasible assignments  $\Omega[i]$ , ensuring that the overall solution constructed thus far given by  $\pi$  remains feasible.

To avoid the risk of the algorithm getting trapped in a cycle, a taboo list containing all the priority sequences evaluated thus far is also maintained. This cycle of analyzing the solution, prioritizing the vessels, critical-shaking the priority sequence and constructing a new solution, is then repeated for a given maximum number of iterations. In our implementation, the algorithm is run for 1000 iterations, and the best solution generated is accepted as the final solution. The complete meta-heuristic is described in Algorithm 5.

---

**Algorithm 5:** CSNS Metaheuristic

---

**Data:** data file,  $\Omega$

**Result:** bestSolution

```

1 begin
2   currentSolution  $\leftarrow$  initialHeuristic( $\Omega$ )
3   bestSolution  $\leftarrow$  current
4   iteration  $\leftarrow$  0
5   tabooList  $\leftarrow$   $\emptyset$ 
6   while iteration < maxIterations do
7     p  $\leftarrow$  assignPriority(currentSolution)
8     p  $\leftarrow$  criticalShake(p)
9     while p  $\in$  tabooList do
10      p  $\leftarrow$  randomize(p)
11     tabooList = tabooList  $\cup$  p
12     currentSolution  $\leftarrow$  construct( $\Omega$ , p)
13     if currentSolution < bestSolution then
14       bestSolution  $\leftarrow$  currentSolution
15     iteration++
16 print bestSolution

```

---

## 4.8 Results and Discussion

In this section we describe the test instances and provide results from the proposed branch-and-price and metaheuristic algorithms.

### 4.8.1 Generation of Instances

The proposed solution algorithms were run on an x64-bit Intel Core i7, (1.60 GHz) using a 64-bit version of CPLEX 12.3. The algorithms were implemented in Java Programming language.

## Chapter 4. The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports

---

### Algorithm 6: Construction Heuristic

---

**Data:**  $\Omega, p$

**Result:**  $\pi$  - constructed solution

```
1 begin
2    $\pi \leftarrow \emptyset$ 
3   for  $i \in p$  do
4      $\text{bestFeasible}[i] \leftarrow \text{findFeasible}(\Omega[i])$ 
5     for  $j \in \Omega[i]$  do
6       if  $j$  is compatible with  $\pi$  &  $j \leq \text{bestFeasible}[i]$  then
7          $\text{bestFeasible}[i] \leftarrow j$ 
8      $\pi = \pi \cup \text{bestFeasible}[i]$ 
9   return  $\pi$ 
```

---

The test instances are based on a sample of data obtained from the SAQR port, Ras-Al-Khaimah, UAE, the biggest bulk port in the middle east. The data was provided for a time horizon of roughly 10 days from 28th March to 6th April, 2011. It further provided information about the physical attributes of the vessels berthing at the port such as the draft and the length of the vessels, expected and actual times of berthing activities including the arrival times and the handling times of the vessels, and the cargo tonnage of the vessels. Based on our observations and notes during our visit to the port in the year 2010, and the data sample provided to us, we could get an estimate of the range of values for most input parameters in our integrated model for berth allocation and yard assignment. Thus given our limited access to real bulk port data, the test instances are designed to be as realistic as possible, to test and validate the effectiveness of the proposed exact algorithm.

Three sets of instances are designed for  $|N|=10$ ,  $|N|=25$  and  $|N|=40$  respectively. In terms of the level of congestion as determined by the inter-arrival times of the incoming vessels, three different types of scenarios are considered in each set of instances - congested, mildly congested and uncongested. In all test instances, the total quay length is 1600 meters, and the vessel lengths lie in the range of 50 to 260 meters as in SAQR. In practice, the time frame of the integrated decision problem of berth allocation and yard assignment is of the order of few days. In the computational study, we have tested instances for the planning horizon of 5 days or 120 hours.

In all the generated instances, the cargo quantities on the vessels vary between 1 and 15 cargo units, and five cargo types are considered - liquid bulk that needs

the pipeline facility for discharging, rock aggregates that need the conveyor facility for loading the vessels and three other general dry cargo types - clay, coal and cement. The incompatible pair of cargo types that cannot be stored at adjacent locations in the yard are liquid bulk - clay, and rock - cement. The berthing layout along with the location of the fixed facilities such as conveyors and pipelines used in the instances is shown in the Table 4.5. As illustrated in Figure 4.2, there are 10 cargo locations in the yard, where location 1 has the pipeline facility and is thus designated for liquid cargo; likewise location 10 has the conveyor facility and is designated to the rock aggregates, while locations 2 to 9 are designated to other cargo types. Thus the assignment of the cargo types to the locations 2 to 9 are decision variables in the optimization problem we are solving. The value of the parameter  $T$  is based on the crane handling rate of 1000 tonnes per hour. The number of cranes operating in a given section is determined by the length of the section, assuming an additional crane for every 50 meters of section length. The parameter  $V_w$  which is a constant dependent on the rate of transfer of cargo is assumed to be equal to 1/1200 hours per meter per unit cargo for the conveyor, 1/3600 hours per meter per unit cargo for the pipeline and 1/600 hours per meter per unit cargo for all other cargo types.

Section	Length	Facility
1	150	C
2	50	C
3	200	P
4	150	C,P
5	125	C,P
6	250	-
7	250	-
8	75	P
9	150	P
10	200	-

Table 4.5: Berthing layout and fixed facility positions for  $|M|=10$  (C and P stand for conveyor and pipeline respectively)

## 4.8.2 Computational Results and Analysis

### Branch-and-Price Results

In Tables 4.6, 4.8 and 4.9, the results obtained from the column generation method for instances containing  $|N|=10, 25$  and 40 vessels respectively are shown. Note that the computation time shown in the table is the time taken to solve

## Chapter 4. The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports

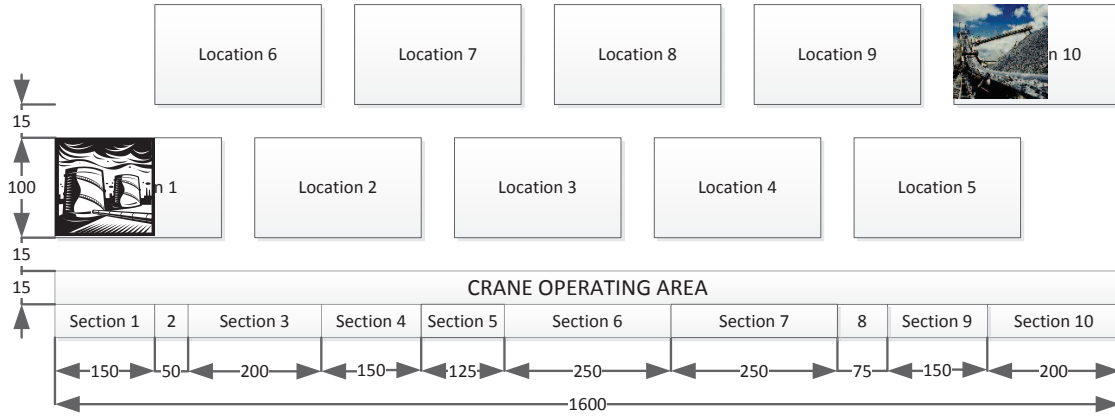


Figure 4.2: Schematic representation of the port used in the test instances

the LP relaxation of the master problem using column generation, while  $\#iter$  indicates the number of iterations of the column generation process to converge to the solution of the LP relaxation. The final size of the active pool of columns is given by  $|\Omega_1|$ , while the total number of feasible assignments is given by  $|\Omega|$ . As discussed earlier, the solution value obtained from the column generation method provides a lower bound to the optimal solution of the original problem. The upper bound to the optimal solution is obtained by running the RMP with integer decision variables in the last stage of the column generation method.

In the tables shown,  $gap1$  represents the optimality gap between the lower bound and the optimal solution, whereas  $gap2$  represents the optimality gap between the optimal solution and the upper bound. The total gap between the lower bound and the upper bound, represented by  $gap$ , is then simply the sum of  $gap1$  and  $gap2$ .

As shown in Table 4.6, for  $|N| = 10$  vessels, the optimal solution is obtained directly from the column generation method for instances 7 and 8. Thus for these instances, there is no need to run the branch and bound algorithm. For instance 3, the computational time of the branch and bound algorithm explodes and the optimal solution remains unknown.

instance #	congestion	time	# iter.	$ \Omega_1 $	$ \Omega $	lb	ub	gap1	gap2	gap
1	yes	4m 16s	15	1 434	76 823	125	127	1.5%	0%	1.5%
2	mild	3m 16s	14	996	71 174	131	134	1.5%	0.7%	2.2%
3	no	1m 47s	11	737	69 824	113	117	–	–	3.4% <sup>2</sup>
4	yes	5m 13s	21	1 294	71 399	130	131	0%	0.8%	0.8%
5	mild	3m 18s	18	1 027	68 909	131	132	0.7%	0%	0.7%
6	no	1m 39s	17	692	67 989	109	111	1.8%	0%	1.8%
7	yes	3m 59s	17	1 311	80 272	126	126	0%	0%	0% <sup>1</sup>
8	mild	2m 37s	13	1 025	77 392	122	122	0%	0%	0% <sup>1</sup>
9	no	1m 29s	9	502	75 912	110	117	1.8%	4.5%	6.3%

<sup>1</sup> Zero optimality gap obtained from the column generation method without running the branch and bound algorithm.

<sup>2</sup> Optimality gap without running the branch and bound algorithm.

Table 4.6: Results obtained from the column generation for the instances containing  $|N| = 10$  vessels.

## Chapter 4. The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports

instance #	time	value	# of nodes	# of $\mu$	# of $\lambda$
1	10m 02s	127	23	22	0
2	12m 21s	133	43	32	10
3	> 4h 00m 00s	–	–	–	–
4	01m 57s	130	5	4	0
5	03m 15s	132	7	6	0
6	03m 12s	111	33	32	0
7	–	–	–	–	–
8	–	–	–	–	–
9	02m 22s	112	29	28	0

Table 4.7: Results obtained from the branch and bound for the instances containing  $|N| = 10$  vessels.

Note that because of the exponential increase in the time complexity of the branch and bound algorithm with increase in the instance size, the branch and bound algorithm is executed only for instances containing  $|N| = 10$  vessels. The results are shown in Table 4.7, and strongly indicate that branching on the  $\mu$  variables is sufficient in almost all cases. Note that for instances containing  $|N| = 25, 40$  vessels, the total optimality gap is still small, and thus we do not execute the branch and bound algorithm and save on the computational time. In these cases, the upper bound is accepted as the final sub-optimal solution to our problem.

Overall from the results of the column generation shown in Tables 4.6, 4.8 and 4.9, we can observe the following: barring a few exceptions, the computational time increases with the level of congestion, as well as with the final size of the active pool of columns and the number of iterations of the column generation process. On the other hand, the total optimality gap as shown in the tables, is smaller for instances with higher level of congestion. This can be attributed to the fact that the total number of feasible solutions is smaller for congested scenarios.

### Meta-heuristic Results

The solution results obtained from the proposed CSNS meta-heuristic are shown in Tables 4.10, 4.11 and 4.12 for  $|N| = 10, 25$  and 40 vessels respectively. The heuristic results are compared with the upper bounds on the optimal solutions obtained from the branch-and-price algorithm. The CSNS and branch-and-price solutions are represented by *ub1* and *ub2* respectively in the tables shown.



#### 4.8. Results and Discussion

instance #	congestion	time	# iter.	$ \Omega_1 $	$ \Omega $	lb	ub	gap
1	yes	5h 03m 13s	60	6 566	196 564	512	529	3.2%
2	mild	2h 18m 38s	38	6 454	190 814	459	477	3.8%
3	no	48m 08s	25	5 300	182 304	413	449	8.0%
4	yes	1h 34m 32s	26	6 868	205 577	520	532	2.3%
5	mild	1h 02m 18s	18	6 051	199 607	469	481	2.5%
6	no	1h 13m 39s	32	5 030	191 427	436	470	7.2%
7	yes	2h 18m 05s	33	5 757	189 619	506	513	1.4%
8	mild	2h 12m 00s	47	6 184	184 129	454	467	2.8%
9	no	40m 00s	28	4 877	177 479	388	412	5.8%

Table 4.8: Results obtained from the column generation for the instances containing  $|N| = 25$  vessels.

instance #	congestion	time	# iter.	$ \Omega_1 $	$ \Omega $	lb	ub	gap
1	yes	3h 12m 15s	35	9 662	300 441	677	698	3.0%
2	mild	1h 52m 46s	31	9 017	295 301	586	608	3.6%
3	no	1h 26m 20s	39	8 174	273 001	457	483	5.4%
4	yes	7h 17m 20s	79	10 743	314 864	664	678	2.0%
5	mild	4h 02m 14s	55	9 771	310 283	581	593	2.0%
6	no	3h 00m 27s	54	8 620	286 524	470	486	3.3%
7	yes	4h 02m 58s	44	8 727	277 169	721	758	4.9%
8	mild	3h 03m 38s	35	8 412	273 011	641	676	5.2%
9	no	1h 20m 27s	33	7 791	251 869	508	532	4.5%

Table 4.9: Results obtained from the column generation for the instances containing  $|N| = 40$  vessels.

## Chapter 4. The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports

instance #	time	ub1	ub2	gap
1	2s	138	127	8.7%
2	2s	135	134	0.7%
3	2s	123	117	5.1%
4	3s	162	131	23.7%
5	3s	132	132	0.0%
6	2s	114	111	2.7%
7	2s	132	126	4.8%
8	2s	123	122	0.8%
9	2s	113	117	-3.5%

Table 4.10: Results obtained from CSNS for the instances containing  $|N| = 10$  vessels.

instance #	time	ub1	ub2	gap
1	21s	628	529	18.3%
2	19s	574	477	20.3%
3	15s	535	449	19.2%
4	17s	588	532	10.5%
5	15s	534	481	11.0%
6	14s	505	470	7.4%
7	22s	604	513	17.7%
8	21s	577	467	23.6%
9	20s	475	412	15.3%

Table 4.11: Results obtained from CSNS for the instances containing  $|N| = 25$  vessels.

It can be seen that the solution gap increases with increase in the instance size, with a significantly large gap for some instances. However the overall performance of the heuristic is reasonably good, with a gap of less than 18% averaged over all test instances, for a computation time of less than one minute.

Thus in practice, the proposed meta-heuristic clearly provides a good alternative to the planner who is faced with the choice of obtaining reasonably good solutions as quickly as possible for a large scale problem, or solve the problem to near optimality in a longer computation time using the branch-and-price algorithm proposed earlier.

instance #	time	ub1	ub2	gap
1	43s	878	698	25.8%
2	43s	814	608	33.9%
3	37s	669	483	38.5%
4	45s	827	678	22.0%
5	44s	752	593	26.8%
6	41s	637	486	31.1%
7	50s	952	758	26.0%
8	49s	903	676	33.6%
9	43s	759	532	42.7%

Table 4.12: Results obtained from CSNS for the instances containing  $|N| = 40$  vessels.

## 4.9 Conclusions and Future Work

In this research, we study the integrated problem of berth allocation and yard assignment in context of bulk ports. In the past, few scholars have attempted to study the berth allocation problem in integration with the yard assignment problem in the context of container terminals, while in the context of bulk ports the problem has not been studied at all. The specific issues in bulk port operations that necessitate the need to devise specific solutions for bulk ports are emphasized in this study. An exact solution algorithm based on branch-and-price and a meta-heuristic based on critical-shaking neighborhood search are proposed to solve the integrated problem of berth allocation and yard assignment. To the best of our knowledge, this is the first study that proposes an exact method to solve this large scale problem in the context of port operations planning, as all the previous studies in the context of container terminals use meta-heuristics to solve the problem.

The mathematical formulation of the integrated berth allocation and yard assignment problem is complex and extensive, however there is still scope for improvement. An important assumption in our model is that each vessel carries only a single type of cargo. In order to model and solve the problem for multiple cargo types, the location of each cargo type on the vessel needs to be explicitly modeled which would require the modification of some of the existing constraints in the current formulation as well as addition of some new constraints. Another possible extension of the current work is to account for the uncertainty in arrival times of the vessels and delays in handling operations due to factors such as breakdown of handling equipment including convey-

## **Chapter 4. The Integrated Berth Allocation and Yard Assignment Problem in Bulk Ports**

---

ors, pipelines and/or mobile harbor cranes etc. In the proposed exact solution method, the major source of the time complexity has been identified as the solution of the sub-problems in the column generation framework . Thus it may be worth investigating the reduction in the solution time of the sub-problems by using dynamic programming and heuristic methods instead of directly using optimization solvers. There is also scope to obtain sub-optimal integer solutions in a more time efficient manner by using heuristic methods (GRASP for instance) instead of using the branch and bound algorithm proposed in this study. Finally more sophisticated techniques such as dual stabilization and dynamic constraint aggregation to speed up the convergence of the column generation process need to be studied and implemented in a way such that they exploit the structure of the specific problem under study.

# Handling Uncertainty **Part II**



## 5 Real-Time Recovery in Berth Allocation

In this chapter, we study the berth allocation problem (BAP) in real time as disruptions occur. In practice, the actual arrival times and handling times of the vessels deviate from their expected or estimated values, which can disrupt the original berthing plan and potentially make it infeasible. We consider a given baseline berthing schedule, and solve the BAP on a rolling planning horizon with the objective to minimize the total realized costs of the updated berthing schedule as the actual arrival and handling time data is revealed in real time. The uncertainty in the data is modeled by making appropriate assumptions about the probability distributions of the uncertain parameters based on past data. We present an optimization based recovery algorithm based on set partitioning method and a smart greedy algorithm to reassign the vessels in the events of disruption. Our research problem derives from the real world issues faced by the SAQR port, Ras Al Khaimah, UAE, where the berthing plans are regularly disrupted owing to a high degree of uncertainty in information. A simulation study is carried out to assess the solution performance and efficiency of the proposed algorithms, in which the baseline schedule is chosen as the solution of the deterministic berth allocation problem without accounting for any uncertainty. Results indicate that the proposed algorithms can significantly reduce the total realized costs of the berthing schedule as compared to the ongoing practice of reassigning vessels at the port.

### 5.1 Introduction

The Berth Allocation Problem (BAP) is one of the most critical and widely studied problems in seaport operations planning. Port operations are affected by a high degree of vessel travel time and handling time uncertainty arising

from weather conditions, mechanical problems, port congestion, demand uncertainty, and other factors. Such uncertainty can make berth allocation planning difficult, and planned schedules are often disrupted. To minimize the impact of such disruptions, plans must be updated dynamically. Most optimization-based approaches for creating berth allocation plans do not explicitly account for uncertainty. Furthermore, the objectives used in such approaches do not explicitly consider objectives useful during re-planning.

Two approaches are used for managing uncertain disruptions in transportation scheduling. In the first approach, systematic robustness is built into the planned, or baseline, schedule. Stochastic optimization models (see Birge and Louveaux (1997), Kall and Mayer (2005) and Wallace and Ziemba (1997)) address problems of this type by minimizing expected operational costs given a probabilistic representation of possible outcomes, while robust optimization models (see Soyster (1973), Bertsimas and Sim (2003), Ben-Tal and Nemirovski (1998), Ben-Tal and Nemirovski (1999), Ben-Tal and Nemirovski (2000) and Bertsimas and Sim (2004)) alternatively focus on worst-case performance over some subset of possible outcomes. The second approach to managing disruptions is to build reactive models for modifying a schedule in real-time in response to new information; it is common to refer to these optimization problems as *recovery problems*. Recovery optimization problems usually use a deterministic information model. To measure the effectiveness of dynamic reactive models, a competitive ratio between the system cost resulting from repeated application of a reactive optimization model and the optimal cost found by *a posteriori* optimization may be computed (Albers (2003)).

In this research, we consider the problem of real-time berth rescheduling. The underlying model is the dynamic, hybrid berth allocation model for bulk ports developed by Umang et al. (2013). We consider uncertainty in both the arrival times and the handling times of the vessels. The objective is to minimize the total realized costs of the modified berthing schedule, which is the sum of the total service cost of the vessels, the rescheduling costs created by altering the berthing times and positions of vessels from a baseline schedule, and the delay to arriving vessels, discussed in more detail later in the chapter.

## 5.2 Literature Review

Comprehensive literature surveys covering operations research approaches to berth allocation problems in container terminals can be found in Bierwirth and



Meisel (2010), Steenken et al. (2004) and Stahlbock and Voss (2008). The deterministic berth allocation problem (BAP) in bulk ports with dynamic vessel arrivals and hybrid berth layout is studied by Umang et al. (2013).

Few studies propose robust planning methods for berth allocation, although there are some recent examples. Zhen, Lee and Chew (2011) use a meta-heuristic approach to solve a two-stage stochastic BAP given a fixed set of scenarios, where the objective is to minimize the total cost of a baseline schedule and the expected cost of recourse. The recourse cost in this study is the weighted time and space deviation of the realistic schedule from the baseline schedule. Han et al. (2010) use a simulation-based genetic algorithm to solve an integrated berth and quay crane scheduling problem with uncertainty in vessel arrival and operation times. For given probability density functions, the objective is to minimize the sum of expected value and standard deviation of the service time and the weighted tardiness of the vessels.

Other papers propose surrogate measures of berth schedule robustness, and incorporate these measures into an optimization objective. For example, Moorthy and Teo (2006) use a sequence pair approach to design a robust berth template for transshipment hubs in container terminals, in which the conflicting objectives are to minimize the total expected delays and deviation from the most preferred berthing locations. Zhen and Chang (2012) define robustness as the weighted sum of the free slack times in the berthing schedule, where weights are determined according to the vessel priorities. A bi-objective model is proposed that minimizes cost and maximizes robustness. Xu et al. (2012) solve a continuous berth allocation problem with uncertainty in vessel arrival and handling times, in which the objective is to balance level of service measured by total vessel departure delay with a robustness measure defined by length of time buffers inserted between vessels occupying the same berthing location to absorb uncertain delays.

Little research has addressed real-time management of berth allocation. In practice, simple rules of thumb guide rescheduling of vessels. Since actual vessel arrival times, and to a lesser extent vessel handling times, may differ substantially from those assumed when developing a baseline schedule, it should be clear that rescheduling will often be required and it is important to do so effectively. A simple but naive approach to rescheduling is to not shift planned vessel berthing positions when rescheduling, and to simply serve vessels at the earliest feasible time given a first-in first-out (FIFO) ordering specified by the planned berthing times. A different approach is to apply an optimization model

in a roll-out procedure for vessel rescheduling, but doing so requires some attention to detail. Zeng et al. (2012) and Du et al. (2010) are a couple of examples of related works. Zeng et al. (2012) address the problem of disruption recovery in the integrated berth and quay crane assignment problem in container terminals. They develop optimization models for re-allocation of berth assignment and rescheduling of quay cranes, but solve the disruption recovery problem using local rescheduling and tabu search methods. Du et al. (2010) use a feedback procedure to develop a robust berth allocation plan and a reactive strategy that takes into account the priorities assigned to the vessels and the congestion at the port.

In this research, we develop a methodology to model the uncertainty in the yet-to-be-revealed arrival times and handling times of the vessels, based on probability distributions derived from past data. We propose a recovery algorithm based on re-optimization of the berthing schedule in the events of disruption and a heuristic based smart greedy algorithm for berth rescheduling in real time. The objective is to minimize the total realized cost of the updated schedule. Our research problem is motivated by challenges faced along these lines at the SAQR port, Ras al-Khaimah, United Arab Emirates, where planned operations are frequently disrupted due to a high degree of uncertainty in the vessel arrival and handling times.

### 5.3 Problem Statement

#### 5.3.1 Baseline Schedule

We study the berth allocation problem in real time for a given baseline schedule. The vessel arrival process is dynamic and stochastic. We assume a hybrid berthing layout and a fixed planning horizon partitioned into discrete time buckets, where each vessel may occupy multiple discrete berth sections, but a given section may be occupied by at most one vessel at a given time, as shown in Figure 5.1.

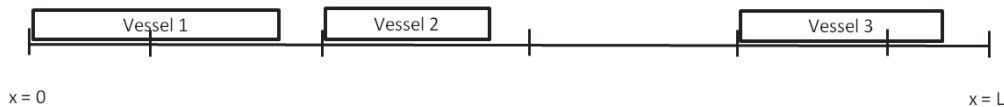


Figure 5.1: *Hybrid berthing layout, showing a feasible assignment of vessels to berth sections at a single point in time*

As demonstrated in Chapter 3, the dynamic hybrid berth allocation problem with known arrival and handling times can be effectively modeled and solved as a generalized set-partitioning problem (GSPP) for relatively large problem size and time horizon of few days. In this approach, the set of all feasible single-vessel berthing assignments is generated *a priori* and is denoted by the set  $P$ . Note that a berthing assignment for a single vessel specifies the berth sections that will be occupied by the vessel, its berthing time, and its completion time (equal to the berthing time plus the handling time). The assignment matrix contains a column for each of the  $|P|$  assignments, and is composed of upper submatrix  $A$  and lower submatrix  $B$ . Each column  $p$  in submatrix  $A$  has a single non-zero value, where row  $i$  contains the value one if the berthing assignment is for vessel  $i \in N$ . Submatrix  $B$  contains a single row for each (berth section, time bucket). Non-zero values in submatrix  $B$  are equal to one if the vessel berthing assignment specified by column  $p$  requires that the vessel occupies the (section, time) represented by the row. To illustrate this idea, consider an example with two vessels 1 and 2 as shown in Figure 5.2. Suppose the quay has 3 discrete berth sections, and that the planning horizon has 3 discrete time periods. Vessel 1 requires specialized handling equipment only available in sections 1 and 2, and vessel 2 arrives at the start of time 2. The resulting assignment matrix is shown in Table 5.1. The first column represents the berthing assignment of vessel 1 to sections 1 and 2 from time 1-2, and so on.

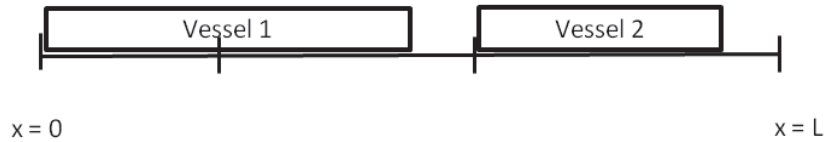


Figure 5.2: Simple example of set partitioning to solve the BAP with  $|N| = 2$ ,  $|M| = 3$  and  $|H| = 3$

We assume the following input data to be available for the GSPP model:

- $H =$  set of discrete time intervals in the planning horizon
- $P =$  set of feasible assignments
- $s = 1, \dots, |H|$  discrete time intervals in the planning horizon
- $p = 1, \dots, |P|$  feasible assignments
- $d_p =$  delay associated with assignment  $p$
- $h_p =$  handling time associated with assignment  $p$

The assignment matrix coefficients are defined as follows.

Vessel 1	1	1	0	0
Vessel 2	0	0	1	1
Section 1, Time 1	1	0	0	0
Section 1, Time 2	1	1	1	0
Section 1, Time 3	0	1	1	0
Section 2, Time 1	1	0	0	0
Section 2, Time 2	1	1	1	1
Section 2, Time 3	0	1	1	1
Section 3, Time 1	0	0	0	0
Section 3, Time 2	0	0	0	1
Section 3, Time 3	0	0	0	1

Table 5.1: Assignment matrix for a simple example of GSPP

$$A_{ip} = \begin{cases} 1 & \text{if the assigned vessel in feasible assignment } p \text{ is vessel } i; \\ 0 & \text{otherwise.} \end{cases}$$

$$b_p^{ks} = \begin{cases} 1 & \text{if section } k \text{ is occupied at time } s \text{ in assignment } p; \\ 0 & \text{otherwise.} \end{cases}$$

There is only a single decision variable in the GSPP model for selection of feasible assignments in the optimal solution which is defined as follows.

$$\lambda_p = \begin{cases} 1 & \text{if assignment } p \text{ is part of the optimal solution;} \\ 0 & \text{otherwise.} \end{cases}$$

The GSPP model is formulated as shown below:

$$\min \sum_p (d_p \lambda_p + h_p \lambda_p) \quad (5.1)$$

$$\text{s.t. } \sum_p (A_{ip} \lambda_p) = 1 \quad \forall i \in N \quad (5.2)$$

$$\sum_p (b_p^{ks} \lambda_p) \leq 1 \quad \forall k \in M, \forall s \in H \quad (5.3)$$

$$\lambda_p \in \{0, 1\} \quad \forall p \in P \quad (5.4)$$

In the deterministic model, the objective (5.1) is to minimize the total service cost of the vessels berthing at the port, which includes the total berthing delays and the total handling cost of the vessels. Constraints (5.2) ensure that each vessel must have exactly one feasible assignment in the optimal solution. Con-

straints (5.3) ensure that a given section at a given time can be occupied by at most one vessel.

### 5.3.2 Real Time Recovery

In practice, the actual arrival and handling times of vessels may deviate from their estimated values, which can disrupt the baseline schedule and possibly render it infeasible. To create a model for berth schedule recovery, we first describe a dynamic information model based on actual seaport operations. We assume that the port receives dynamic updates on the estimated arrival time of each inbound vessel. Suppose that these estimated arrival time updates occur sporadically for each vessel, and that each update occurs before the actual arrival of the vessel at time  $a_i$ . We assume that  $a_i$  is known with certainty only at time  $a_i$ . In the case of handling time, we assume that a single estimate of handling time is known in advance and that actual handling time  $h_i$  of vessel  $i$  is only known when handling is completed.

To design an optimization problem for schedule recovery, suppose that re-planning is initiated at time  $t$  for a given baseline schedule. At this time, we can partition the vessel set  $N$  into five subsets as follows:

$$N = N_1^t \cup N_2^t \cup N_3^t \cup N_4^t \cup N_5^t$$

where,

- $N_1^t$  is the subset of vessels which have been berthed, completely served, and have departed the port;
- $N_2^t$  is the subset of vessels which are currently berthed, at known berthing locations that cannot be altered;
- $N_3^t$  is the subset of vessels which arrived to the port, but have not yet been berthed;
- $N_4^t$  is the subset of vessels which have not arrived yet, but that have an estimated arrival time  $\hat{a}_i^t > t$ ; and
- $N_5^t$  is the subset of vessels which have not arrived yet, but that have an estimated arrival time  $\hat{a}_i^t \leq t$ .

## Chapter 5. Real-Time Recovery in Berth Allocation

Vessel	Expected Arrival Time	Baseline Berthing Time	Expected Handling Time
1	0	0	3
2	0	1	2
3	0	1	2

Table 5.2: Baseline schedule for illustrative example with  $|N|=3$ ,  $|M|=6$  and  $|H|=4$

Note that the set of unassigned vessels at time instant  $t$  is  $N_u^t = N_3^t \cup N_4^t \cup N_5^t$ .

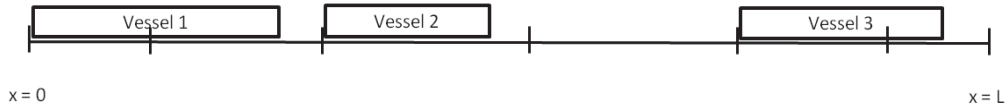


Figure 5.3: Baseline berthing positions for illustrative example with  $|N|=3$ ,  $|M|=6$  and  $|H|=4$

Consider the example containing three vessels 1, 2 and 3 as shown in Figure 5.3. The berthing positions of the vessels in the original baseline schedule are as shown in the figure. We consider 6 quay sections and 4 discrete time intervals in the planning horizon. The arrival and handling time information related to all the vessels is given in Table 5.2

Time	Vessel	Arrival Time Updated?	Vessel Arrived	Vessel Assigned	Vessel Completed	Subset
t=0	1	Yes $\rightarrow$ 1	No	No	No	$N_4^t$
	2	Yes $\rightarrow$ 1	No	No	No	$N_4^t$
	3	-	Yes	No	No	$N_3^t$
t=1	1	Yes $\rightarrow$ 2	No	No	No	$N_4^t$
	2	No	No	No	No	$N_5^t$
	3	-	Yes	Yes	No	$N_2^t$
t=2	1	-	Yes	Yes	No	$N_2^t$
	2	-	Yes	Yes	No	$N_2^t$
	3	-	Yes	Yes	No	$N_2^t$
t=3	1	-	Yes	Yes	No	$N_2^t$
	2	-	Yes	Yes	No	$N_2^t$
	3	-	Yes	Yes	Yes	$N_1^t$
t=4	1	-	Yes	Yes	Yes	$N_1^t$
	2	-	Yes	Yes	Yes	$N_1^t$
	3	-	Yes	Yes	Yes	$N_1^t$

Table 5.3: Disruption Scenario for a simple example with  $|N|=3$ ,  $|M|=6$  and  $|H|=4$

Now consider the disruption scenario detailed in Table 5.3. Each large row in the table describes the events that have occurred by time  $t$ . Note that the arrival time of vessel 1 is updated twice, and it actually arrives at time 2. Although the expected handling time for vessel 1 is 3, actual handling time is 2 and the vessel departs at time 4. The arrival time of vessel 2 is updated once, and the actual handling time is equal to the expected value of 2. Note that at time  $t=1$ , vessel 2 belongs to the subset  $N_5^t$ , since there is no available information about the future expected arrival time of the vessel. For vessel 3, the actual arrival and handling times are the same as the expected values and there are no arrival information updates.

Once a berthing schedule is determined, the terminal manager may begin allocating resources such as cargo storage facilities, labor, and handling equipment according to the requirements of the berthing vessels. When the baseline schedule is disrupted, the manager must reallocate these resources and incur additional costs. Thus, an important objective in berthing schedule recovery is to minimize the deviation of the realized berthing plan from the baseline. Another important measure is the fairness of any required rescheduling. Here, we attempt to ensure fairness by prioritizing the minimization of berthing delay for vessels that arrive on time. Finally, it is also important to maximize overall port productivity which we measure using the total actual flow time of all berthing vessels in the usual way.

We propose therefore the following optimization model for real time recovery of a baseline berth schedule. The model is to be solved given a baseline schedule at time epoch  $t$ , and considers only a single scenario for all uncertain parameters (to be described in more detail in Section 5.4.1).



The input parameters for the model are:

- $|H|$  = duration of planning horizon
- $L$  = total length of quay
- $M$  = set of berth sections
- $N_a^t$  = subset of active vessels at time  $t$  that have not yet departed the port
- $N_u^t$  = subset of vessels at time  $t$  that have not been assigned to a berth position
- $N_o^t$  = subset of unassigned vessels at time  $t$  that arrived on-time or are expected to arrive on-time
- $A_i$  = planned arrival time of vessel  $i$
- $\mu_i$  = service priority of vessel  $i$
- $a_i^t$  = (updated) arrival time of vessel  $i$  at time  $t$
- $b_{ik}$  = 1 if baseline berthing location  $k \in M$  used for vessel  $i$  (starting berth section), 0 otherwise
- $m_i$  = baseline berthing time of vessel  $i$
- $e_i$  = baseline departure time of vessel  $i$
- $g_k$  = linear coordinate of berthing location  $k$
- $L_i$  = length of vessel  $i$
- $h_{ik}^t$  = (updated) handling time at time  $t$  for vessel  $i$  berthed at starting berth section  $k$
- $M_i \subseteq M$ , subset of starting berth sections for which vessel can be feasibly berthed due to draft and length restrictions
- $c_1$  = penalty cost of shifting a vessel by a unit distance along the quay
- $c_2$  = penalty cost of unit delay time beyond baseline departure time for a vessel
- $c_3$  = penalty cost of unit delay time beyond baseline service time for a vessel arriving on-time

The decision variables are:

- $m_i' \geq 0$  updated berthing time of vessel  $i$
- $e_i' \geq 0$  updated departure time of the vessel  $i$
- $w_i' \geq 0$  difference between the updated service time and the estimated service time as per the baseline schedule of the vessel  $i$
- $b_{ik}'$  binary, equals 1 if vessel  $i$  updated berthing location is  $k$ , 0 otherwise
- $y_{ij}$  binary, equals 1 if vessel  $i$  berthed to the left of vessel  $j$  along quay, 0 otherwise
- $z_{ij}$  binary, equals 1 if vessel  $i$  departs no later than the berthing time of vessel  $j$ , 0 otherwise

The optimization model can be formulated as follows:

$$\min Z_t = Z_{1t} + Z_{2t} + Z_{3t} \quad (5.5)$$

$$Z_{1t} = \sum_{i \in N_u^t} (e'_i - a_i^t) \quad (5.6)$$

$$Z_{2t} = \sum_{i \in N_u^t} \left( c_1 \left| \sum_{k \in M} g_k b'_{ik} - \sum_{k \in M} g_k b_{ik} \right| + c_2 \mu_i |e'_i - e_i| \right) \quad (5.7)$$

$$Z_{3t} = \sum_{i \in N_o^t} c_3 w'_i \quad (5.8)$$

subject to the constraints

$$m'_i - a_i^t \geq 0 \quad \forall i \in N_u^t \quad (5.9)$$

$$e'_i - m'_i - \sum_{k \in M} h_{ik}^t b'_{ik} = 0 \quad \forall i \in N_a^t \quad (5.10)$$

$$w'_i \geq (e'_i - a_i^t) - (e_i - A_i) \quad \forall i \in N_o^t \quad (5.11)$$

$$\sum_{k \in M} (g_k b'_{jk}) + B(1 - y_{ij}) \geq \sum_{k \in M} (g_k b'_{ik}) + L_i \quad \forall i \in N_a^t, j \in N_u^t, i \neq j \quad (5.12)$$

$$m'_j + B(1 - z_{ij}) \geq m'_i + \sum_{k \in M} h_{ik}^t b'_{ik} \quad \forall i \in N_a^t, j \in N_u^t, i \neq j \quad (5.13)$$

$$y_{ij} + y_{ji} + z_{ij} + z_{ji} \geq 1 \quad \forall i \in N_a^t, j \in N_u^t, i \neq j \quad (5.14)$$

$$\sum_{k \in M_i} b'_{ik} = 1 \quad \forall i \in N_u^t \quad (5.15)$$

$$b'_{ik} \in \{0, 1\} \quad \forall i \in N_a^t, \forall k \in M \quad (5.16)$$

$$y_{ij} \in \{0, 1\} \quad \forall i \in N_a^t, j \in N_u^t \quad (5.17)$$

$$z_{ij} \in \{0, 1\} \quad \forall i \in N_a^t, j \in N_u^t \quad (5.18)$$

The equations (5.5)-(5.8) minimize the total realized cost  $Z_t$  at time instant  $t$ , which is the sum of the total service cost of the vessels given by  $Z_{1t}$ , the total cost of rescheduling the vessels given by  $Z_{2t}$  and the delays beyond the esti-

mated service times of vessels arriving on-time given by  $Z_{3t}$ . Note that while the parameters  $\mu_i$  are dependent on the relative priorities assigned by the port authority to the vessels berthing at the port, the choice of the parameters  $c_1$ ,  $c_2$  and  $c_3$  needs to be adapted in accordance with the actual real cost incurred by the port due to each of the cost components under different disruption scenarios. In the absence of such data, the parameter values were chosen on the basis of intuition and trials in our study. Constraints (5.9) are the dynamic arrival constraints. Constraints (5.10) state that vessels depart as soon as their processing is finished. Note that in constraints (5.10),  $m'_i$  and  $b'_{ik}$  are preset and unchangeable for vessels that have already been assigned to a berth position. Constraints (5.11) are used to define the variables  $w'_i$  as the difference in the updated service time and the estimated service time of the vessel  $i$  as per the baseline schedule. Constraints (5.12)-(5.14) are the non-overlapping restrictions for any two vessels berthing at the port. Constraints (5.15) ensure that each vessel has exactly one starting berth section, and occupies sections for which the vessel can be feasibly berthed due to length and draft restrictions. Note that the vessel arrival times and handling times in the objective function term (5.6) and the constraints (5.9), (5.10), (5.11) and (5.13) do not represent the actual values, but our expectation of the arrival and handling time values at time instant  $t$ , which may or may not have been revealed up to that time instant. The modeling of uncertainty in the arrival times and handling times of the vessels is discussed in the next section.

## 5.4 Recovery Methodology

### 5.4.1 Modeling Uncertainty

We propose that recovery decisions to be determined via optimization over a rolling planning horizon. At any given time  $t$ , certain vessel arrival and handling times are known with certainty, while other information is not known.

#### Uncertainty in arrival times

We use a simple approach for modeling uncertain arrival times. Based on sample data from the port, we assume that vessel arrival times are uniformly distributed around the expected arrival time. Specifically, for vessel  $i \in N$ , the actual arrival time  $a_i$  lies in the interval  $[A_i - V, A_i + V]$  where  $A_i$  is the expected arrival time. When planning, we assume that the most recent update  $a_i^t$  is appropriate for use during planning. Then, we model arrival times at time  $t$  as

follows:

- If vessel  $i \in N_2^t \cup N_3^t$ , implying that the vessel has actually arrived, then  $a_i^t$  is known and equal to  $a_i$ ;
- If vessel  $i \in N_4^t$ , then  $a_i^t$  is assumed equal to the last update of its arrival time. Thus if the updated arrival time of the vessel at time  $t_1 \leq t$  was  $a_i^{t_1}$  and no other arrival time update occurs between  $t_1$  and  $t$ , then the planned arrival time  $a_i^t$  is assumed equal to  $a_i^{t_1}$ ; and
- If vessel  $i \in N_5^t$ , then the most recent vessel arrival time update was inaccurate and we therefore assume that the vessel may arrive any time on the interval  $[t, A_i + V]$ . We use a planning time  $a_i^t$  where

$$\text{Prob}(a_i \leq a_i^t) = \rho_a$$

where  $\rho_a$  is an input probability and  $a_i$  is assumed to be uniformly distributed on  $[t, A_i + V]$ , and thus

$$a_i^t = t + \rho_a(A_i + V - t)$$

Note that  $1 - \rho_a$  can be interpreted as a likelihood of the infeasibility of the schedule determined at time instant  $t$  due to late arrival of vessels in  $N_5^t$ .

### Uncertainty in handling times

Based on sample data from the port, the handling times of vessels are modeled using truncated exponential distributions. In practice, actual handling times are usually close to estimated values, but in cases of equipment breakdown or other mechanical problems the times may be significantly longer. For any vessel  $i \in N$  berthed at the starting section  $k \in M$ , the handling time is assumed to be distributed according to a truncated exponential distribution on the interval  $[H_{ik}, \gamma H_{ik}]$  where  $H_{ik}$  is the minimum handling time and  $\gamma \geq 1$  is a factor used to define an upper bound. We then use the following two cases to determine a planned handling time  $h_{ik}^t$  at time  $t$ :

- If vessel  $i \in N_2^t$ , then its actual handling time is not yet known but its berth position  $k$  is known. Let  $g_{it}$  be the elapsed processing time of vessel  $i$  at time  $t$ . Then, the total handling time  $h_{ik}$  is assumed to be distributed according to a truncated exponential distribution on the interval  $[\max(g_{it}, H_{ik}), \gamma H_{ik}]$ , and  $h_{ik}^t$  is determined such that

$$\text{Prob}(h_{ik} \leq h_{ik}^t) = \rho_h$$

where  $\rho_h$  is an input probability. Again,  $1 - \rho_h$  is the likelihood that the vessel handling time, realized at time  $t$ , exceeds  $h_{ik}^t$  and thus potentially invalidates the replanned schedule.

- If vessel  $i \in N_3^t \cup N_4^t \cup N_5^t$ , then neither the actual handling time of the vessel nor the actual berthing position of the vessel are yet known. In this case,  $h_{ik}$  is assumed to be distributed according to a truncated exponential distribution on the interval  $[H_{ik}, \gamma H_{ik}]$ , and  $h_{ik}^t$  is determined using the same expression

$$\text{Prob}(h_{ik} \leq h_{ik}^t) = \rho_h$$

Note that in these cases, the expected handling time of vessel  $i$  at time  $t$  berthed at position  $k$  is given by

$$h_{ik}^t = -(1/\tau) \ln(e^{-\tau L_{ik}^t} - \rho_h(e^{-\tau L_{ik}^t} - e^{-\tau U_{ik}^t}))$$

where  $L_{ik}^t$  and  $U_{ik}^t$  are the left and right extremes of the discrete truncated exponential distribution of the handling time of the vessel at time  $t$ , and  $\tau$  is the parameter of the distribution.

### 5.4.2 Solution Algorithms

#### Traditional Greedy Algorithm

In this section, we briefly discuss the current practice for reassigning vessels given a baseline schedule at the port. We call the corresponding algorithm the *traditional greedy algorithm* for schedule recovery. In this approach, we move forward in time from present and assign each incoming vessel to the berthing location where the cost of reassigning the vessel is minimized. The cost of reassignment includes all the three cost components associated with the particular vessel to be reassigned. A vessel is assigned to a berthing location as soon as space is available, but not before the planned berthing time in the original baseline schedule. The implementation is presented in detail in Algorithm 7.

---

**Algorithm 7:** Algorithm for implementation of traditional greedy based recovery algorithm to solve the BAP in real time

---

**Require:** Baseline schedule for set  $N$  of vessels, set  $M$  of sections

```
while time  $\leq |H|$  do
  for Berthing Schedule:  $b$  do
    if  $b.hasArrived$  AND  $!b.isAssigned$  then
      for  $k = 1 \rightarrow M$  do
        if  $isStartSectionAvailable(b.vessel, k)$  AND  $cost(b.vessel, k) <$ 
           $minimumCost$  then
           $foundSection = true;$ 
           $minimumCost = cost(b.vessel, k)$ 
           $assigned\_start\_section = k$ 
        end if
      end for
      if  $foundSection$  AND  $time \geq b.estimatedBerthingTime$  then
         $Assign(b.vessel, assigned\_start\_section)$ 
      end if
    end if
  end for
   $time++$ 
end while
```

---

### Optimization Based Recovery Algorithm

The proposed optimization methodology for schedule recovery in real time seeks to re-plan berthing assignments for all vessels not currently berthed at each decision epoch given new information. Re-planning is only required when:

- the arrival time of any vessel is updated, and it deviates from its previous value
- the actual handling time of any vessel is revealed and is not equal to its estimated value

In a given optimization run, the new berthing assignment is determined by the re-optimization of all the unassigned vessels in the schedule at that time instant with the objective function (5.19). The uncertainty in the yet-to-be-revealed arrival times and handling times of the vessels is modeled as described in section 5.4.1. When solving the problem at time instant  $t$  for the rolling planning horizon  $[t, t + H]$ , the berthing assignment of all the vessels whose processing has already started is considered frozen and unchangeable. To prevent space overlapping with vessels which are being currently processed, the occupied sections are blocked for the worst handling time for each of the berthed vessels until their actual handling time is revealed.

The algorithm to reschedule the vessels is implemented by reformulating the optimization model (5.5)-(5.18) as a set-partitioning problem by generating all the feasible assignments of the unassigned vessels in the schedule every time there is a disruption. In the optimization run at time instant  $t$ , the objective function is:

$$\min Z_t = Z_{1t} + Z_{2t} + Z_{3t} \quad (5.19)$$

$$Z_{1t} = \sum_{p \in P_u^t} (d_p^t \lambda_p + h_p^t(k') \lambda_p) \quad (5.20)$$

$$Z_{2t} = \sum_{p \in P_u^t} (c_1 |b_p(k') - b_p(k)| \lambda_p + c_2 \mu_p |e_p' - e_p| \lambda_p) \quad (5.21)$$

$$Z_{3t} = \sum_{p \in P_o^t} (c_3 w_p' \lambda_p) \quad (5.22)$$

subject to the constraints

$$\sum_p (A_{ip} \lambda_p) = 1 \quad \forall i \in N_u^t \quad (5.23)$$

$$\sum_p (b_p^{ks} \lambda_p) \leq 1 \quad \forall k \in M, \forall s \in [t, t + H] \quad (5.24)$$

$$\lambda_p \in \{0, 1\} \quad \forall p \in P_u^t \quad (5.25)$$

Note that the above model is an extension of the set partitioning model (5.1)-(5.4) to solve the deterministic berth allocation problem, including two additional cost terms in the objective function. These two terms, represented by the equations (5.7)-(5.8) in the initial optimization model, are related to the minimization of the weighted space and time deviation of the realized schedule from the baseline schedule, and the delays to vessels arriving on-time respectively.

In the above formulation, the following input data is assumed to be available:



---

$H =$	set of discrete time intervals in the planning horizon
$P_u^t =$	set of feasible assignments of the unassigned vessels at time instant $t$
$P_o^t =$	$\subset P_u^t$ , set of feasible assignments of the unassigned vessels at time instant $t$ that have or are expected to arrive on-time
$d_p^t =$	berthing delay for the vessel estimated at time instant $t$ represented by the assignment $p$
$h_p^t(k') =$	handling time of the vessel estimated at time instant $t$ represented by the assignment $p$ berthed at the starting section $k' \in M$
$b_p(k) =$	estimated berthing location as per the baseline schedule of the vessel represented by the assignment $p$
$e_p =$	estimated departure time as per the baseline schedule of the vessel represented by the assignment $p$
$b_p(k') =$	updated berthing location of the vessel represented by assignment $p$
$e'_p =$	updated departure time of the vessel represented by assignment $p$
$w'_p =$	updated time difference between the actual service time and the estimated service time as per the baseline schedule of the vessel represented by assignment $p$
$\mu_p =$	service priority assigned to the vessel represented by the assignment $p$
$c_1 =$	cost of shifting the vessel by unit distance along the quay
$c_2 =$	cost of one unit time of delay beyond the departure time of the vessel as per the baseline schedule
$c_3 =$	cost of one unit time of additional berthing delay to a vessel arriving on-time with respect to the baseline schedule
$A_{ip} =$	$\begin{cases} 1 & \text{if assignment } p \text{ is a feasible assignment for vessel } i; \\ 0 & \text{otherwise.} \end{cases}$
$b_p^{ks} =$	$\begin{cases} 1 & \text{if section } k \text{ is occupied at time } s \text{ in assignment } p; \\ 0 & \text{otherwise.} \end{cases}$

There is only a single type of decision variable used in the model for the selection of the feasible assignments in the optimal solution which is defined as follows:

$$\lambda_p = \begin{cases} 1 & \text{if assignment } p \text{ is part of the optimal solution;} \\ 0 & \text{otherwise.} \end{cases}$$

Constraints (5.23) ensure that each unassigned vessel has exactly one feasible assignment in the optimal solution. Constraints (5.24) ensure that a given section at a given time can be occupied by at most one vessel. The implementation

of the optimization based recovery algorithm is described by Algorithm 8. Note that the berthing assignment of a given vessel may be updated several times during the schedule recovery process, but once the handling of the vessel has actually started, it's berthing assignment does not change again thereafter.

---

**Algorithm 8:** Algorithm for implementation of optimization based recovery algorithm to solve BAP in real time

---

**Require:** Baseline schedule of set  $N$  of vessels, set  $M$  of sections

```
Initialize set  $N_u$  of unassigned vessels  $\leftarrow N$ 
Initialize time = 0
while  $|N_u| > 0$  AND time  $\leq |H|$  do
    boolean shouldOptimize  $\leftarrow$  false
    if scheduleDisrupted then
        shouldOptimize = true
    end if
    if shouldOptimize then
        Re-optimize  $\forall i \in N_u$ 
    end if
    for berthing Schedule:  $b$  do
        if ! $b.isAssigned$  AND  $b.hasArrived$  AND counter  $\geq$ 
             $b.estimatedStartTime$  then
            Assign ( $b.vessel$ ,  $b.estimatedStartSection$ )
             $N_u \leftarrow N_u - \{i\}$ 
        end if
    end for
    time++
end while
```

---

### Smart Greedy Algorithm

---

**Algorithm 9:** Algorithm for implementation of smart greedy recovery algorithm to solve the BAP in real time

---

**Require:** Baseline schedule for set  $N$  of vessels, set  $M$  of sections

---

```

while time  $\leq |H|$  do
  for Berthing Schedule:  $b$  do
    if  $b.hasArrived$  AND  $\neg b.isAssigned$  then
      for  $k = 1 \rightarrow M$  do
        if  $isStartSectionAvailable(b.vessel, k)$  AND
           $smartGreedyCost(b.vessel, k) < minimumCost$  then
           $foundSection = true;$ 
           $minimumCost = smartGreedyCost(b.vessel, k)$ 
           $assigned\_start\_section = k$ 
        end if
      end for
      if  $foundSection$  AND  $time \geq b.estimatedBerthingTime$  then
         $Assign(b.vessel, assigned\_start\_section)$ 
      end if
    end if
  end for
   $time++$ 
end while

```

---

In the smart greedy algorithm, the decision to reschedule a particular vessel is based on the cost of the reassignment of the vessel on the whole schedule by modeling the unknown arrival times and handling times of the vessels as described in section 5.4.1. In this approach, every time there is an incoming vessel arriving at the port we scan the entire quay and assign it to the set of sections where the total cost of assignment of all the unassigned vessels at that time instant given by equations (5.5)-(5.8) is minimized. As in the traditional greedy method, the assignment of any incoming vessel is done as soon as berthing space is available for the vessel, but not before the planned berthing time as per the original baseline schedule. In determining the total cost to assign a given vessel at a given set of section(s),

- the arrival times and handling times of all the other unassigned vessels are modeled as described in section 5.4.1
- all the other unassigned vessels are assigned to their estimated berthing sections as per the baseline schedule

- the handling of any unassigned vessel cannot start before the estimated berthing time of the vessel as per the original baseline schedule

The smart greedy recovery algorithm is described in Algorithm 9. It should be noted that unlike the optimization based recovery method that is based on re-optimization in the event of a disruption and updating the schedule, the smart greedy method is based on reassigning a single vessel at a given time and adhering to the original baseline schedule as far as possible.

### A posteriori Optimization

If the problem of recovering a planned berthing schedule in real time for a given time horizon is re-solved after all the actual arrival and handling time information has been revealed, then the problem of real time recovery reduces to solving the deterministic berth allocation problem with the following objective function cost:

$$\min Z = Z_1 + Z_2 + Z_3 \quad (5.26)$$

$$Z_1 = \sum_{p \in P} (d_p \lambda_p + h_p(k') \lambda_p) \quad (5.27)$$

$$Z_2 = \sum_{p \in P} (c_1 |b_p(k') - b_p(k)| \lambda_p + c_2 \mu_p |e'_p - e_p| \lambda_p) \quad (5.28)$$

$$Z_3 = \sum_{p \in P} (c_3 w'_p \lambda_p) \quad (5.29)$$

subject to the constraints

$$\sum_p (A_{ip} \lambda_p) = 1 \quad \forall i \in N \quad (5.30)$$

$$\sum_p (b_p^{ks} \lambda_p) \leq 1 \quad \forall k \in M, \forall s \in [t, t + H] \quad (5.31)$$

$$\lambda_p \in \{0, 1\} \quad \forall p \in P \quad (5.32)$$

Note that in the above formulation, the index  $t$  has been dropped from the variables used in the earlier formulation (5.19)-(5.25). A posteriori optimization is useful to test and validate the solution performance of the proposed recovery algorithms, since the solution to the above formulation is a lower bound to the problem of berth rescheduling in real time. Thus it is used as a benchmark for the comparison of the solution performance of the algorithms.

## **5.5 Results and Analysis**

In this section, we compare the solution performance of the recovery algorithms discussed in the previous section. The algorithms were implemented in JAVA programming language and all tests were run on an Intel Core i7 (2.80 GHz) processor and used a 32-bit version of CPLEX 12.2.

### **5.5.1 Generation of Instances**

In the computational study, the baseline schedule is estimated by solving the deterministic berth allocation problem to optimality, based on instances inspired from real data obtained from SAQR port, Ras Al Khaimah, UAE. The data sample received from the port provided information about the physical attributes of the vessels such as the length and the draft of the vessels, expected and actual times of arrival, berthing times, processing and departure times of vessels, expected and actual berthing positions and the cargo tonnage of the vessels. The data was provided for over 20 vessels for a time horizon of roughly 10 days from 28th March to 6th April, 2011. Based on the data sample and our notes and observations during our visit to the port, we could get an estimate of the range of values for most input parameters in our model.

The relative solution performance of the recovery algorithms is assessed by carrying out a simulation study in which the baseline schedule is subjected to 100 disruption scenarios and the total realized cost of the modified schedule is computed using each recovery method for each simulation run. In our study, the baseline schedule is a combination of cycles of mild or high congestion at the port, as determined by the number of vessels berthing in each cycle. The two baseline schedules considered in the computational study are shown in Figures 5.4-5.5. The length of each cycle  $|H|$  is equal to 5 days or 120 hours. In a period of mild congestion, the number of scheduled vessel arrivals is 10, while in a period of high congestion it is 25. As done in practice, the problem of updating

## Chapter 5. Real-Time Recovery in Berth Allocation

Vessel	ETA	Arrival Updates	ATA
Vessel 0	19	22(2) 21(4) 24(5) 22(6) 24(7) 23(8) 23(9) 23(22)	23
Vessel 1	3		6
Vessel 2	4	7(3) 6(4) 6(5)	7
Vessel 3	14	16(2) 10(3) 12(4)	11
Vessel 4	18	23(9)	22
Vessel 5	12	13(7)	12
Vessel 6	0	5(2)	4
Vessel 7	0		-4
Vessel 8	0		3
Vessel 9	11		7

Table 5.4: A sample arrival disruption scenario for  $|N|=10$  vessels. ETA and ATA stand for the expected and actual arrival times respectively. In the arrival updates, the numbers indicate the updated arrival times and the numbers in the parantheses indicate the time instants at which the updates are received.

the baseline schedule in real time is solved on a rolling planning horizon, where at any given time instant  $t$ , the planning window from  $t$  to  $t+H$  is 120 hours. It is further assumed that the port is empty before time  $t=0$ , and the actual arrival times of the incoming vessels are updated at or after this time. A sample arrival disruption scenario is shown in Table 5.4.

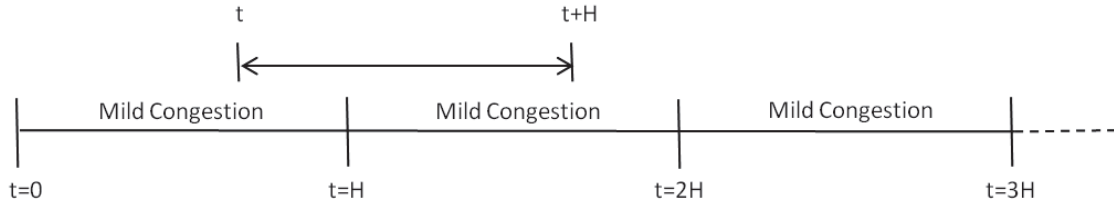


Figure 5.4: Baseline schedule representing the mildly congested scenario

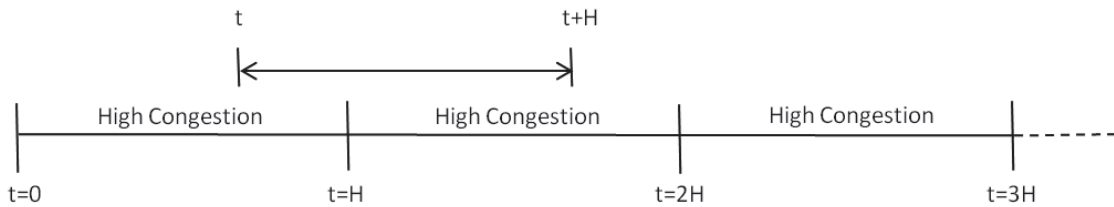


Figure 5.5: Baseline schedule representing the highly congested scenario

Based on the degree of stochasticity with respect to the deviation in the actual arrival and handling times from the estimated values, two types of disruption scenarios are considered in the simulation study:

- Low Stochasticity  
Arrival Time Scenarios:  $V=5$   
Handling Time Scenarios:  $\gamma = 1.1, \tau = 0.5$
- High Stochasticity  
Arrival Time Scenarios:  $V=10$   
Handling Time Scenarios:  $\gamma = 1.2, \tau = 0.5$

The other parameters of the recovery algorithms are selected on the basis of intuition or by trials and are listed as follows:

- On-time arrival:  $U = 4$ .
- Uncertainty parameters:  $\rho_a = \rho_h = 0.95$ .

We conduct two sets of computational experiments based on the inclusion or exclusion of the cost component related to the deviation of the realized schedule from the original baseline schedule. The weight constants in the objective function terms (5.7)-(5.8) are selected as follows:

- In the first set of experiments, the second cost component  $Z_{2t}$  is neglected, implying that the parameters  $c_1$  and  $c_2$  in equation 5.7 are assumed equal to 0. The parameter  $c_3$  in equation 5.8 is assumed equal to 1.
- In the second set of experiments, all the three cost components are considered. The parameters  $c_1$  and  $c_3$  are assumed equal to 1, while the parameter  $c_2$  is assumed equal to 0.002, implying that the cost of shifting a vessel by 500 meters along the quay is considered equivalent to one hour of additional delay.

### 5.5.2 Comparison of Algorithms

#### Excluding Cost of Deviation from the Original Schedule

In Figures 5.6-5.9, results obtained from the simulation study are shown using box plots for the first set of experiments in which the cost of deviation from the original baseline schedule is neglected in the objective function. It can be seen that the optimization based recovery algorithm and the heuristic based

smart greedy recovery method clearly outperform the traditional greedy recovery method for all the four scenarios. Thus if implemented, the proposed algorithms can lead to substantial cost savings to the port.

The optimization based recovery algorithm was found to be better in terms of solution performance than the smart greedy recovery method in most of the tested scenarios. In fact for the mildly congested case, the optimization based method was found to be superior to the other two methods in all the 100 simulation runs. On the other hand, in the high congested case, the performance of the optimization based method was found to be the best in 69% and 51% of the simulation runs for the low and high stochasticity scenarios respectively. However note that while the optimization based recovery algorithm outperforms the smart greedy method in terms of solution performance, it is also computationally more expensive as it may take up to a few minutes to run a single re-optimization as compared to the smart greedy method where the output is returned almost instantaneously. Moreover since the growth in the number of variables and constraints is very fast in the set-partitioning method, the solver can run out of memory for large instance size as defined by the number of vessels, number of sections along the quay and the length of the planning horizon in the problem instance.

Other key performance indicators to assess the solution performance of the proposed algorithms are shown in the Tables 5.5-5.6. The percentage difference in the mean total cost indicates the percentage difference in the mean total objective function cost averaged over the 100 simulations between the recovery algorithm and the solution obtained from the a posteriori optimization method. It can be seen that the solution gap increases with both the level of congestion and the degree of stochasticity. The number of unserved vessels at time  $t=H$  indicates the count of vessels that are scheduled to arrive between  $t=0$  and  $t=H$  and have not left the port at time  $t=H$ . Similarly the number of unserved vessels at time  $t=2H$  indicates the count of vessels that are scheduled to arrive between  $t=0$  and  $t=2H$  and have not left the port at time  $t=2H$ . It can be seen that the optimization based algorithm is superior in terms of both the number of unserved vessels and the average waiting time per vessel. It is interesting to note the significant rise in the average waiting time per vessel expressed in hours with increase in the level of congestion at the port. This result is consistent with our observations during our visit to the port, where the waiting times of few vessels invariably escalate to the order of few days in periods of high congestion and/or stochasticity. The cost of deviation is the objective function cost term in the equation 5.7 with  $c_1$  and  $c_2$  equal to 1 and 0.002 respectively,



which is inversely related to the measure of adherence of the realized schedule to the original baseline schedule. Interestingly, in terms of the adherence to the original schedule, the smart greedy method is superior to the other two recovery methods in all the four scenarios. Thus if a key objective of the port is disruption management i.e. minimization of the deviation from the originally planned schedule to minimize the reallocation of resources, the smart greedy method may be the preferred approach to react to disruptions.

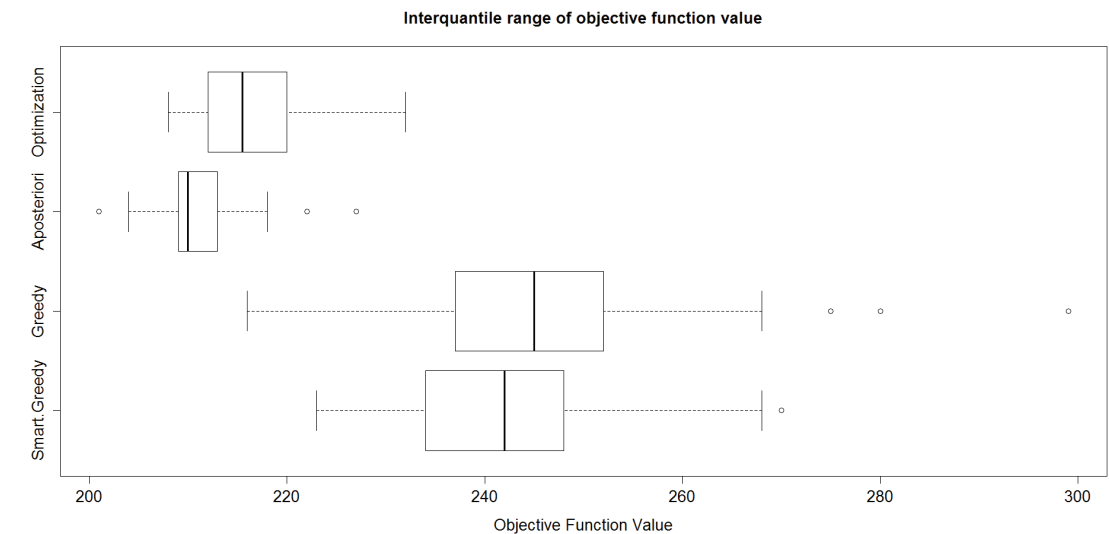


Figure 5.6: Comparison of the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between  $t=H$  and  $t=2H$  for the scenario with mild congestion and low stochasticity

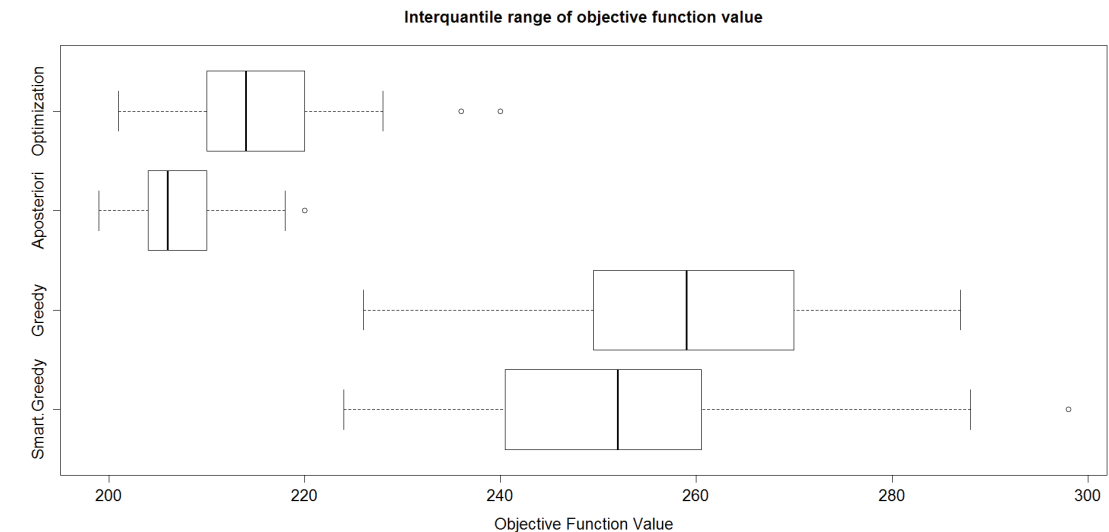


Figure 5.7: Comparison of the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between  $t=H$  and  $t=2H$  for the scenario with mild congestion and high stochasticity

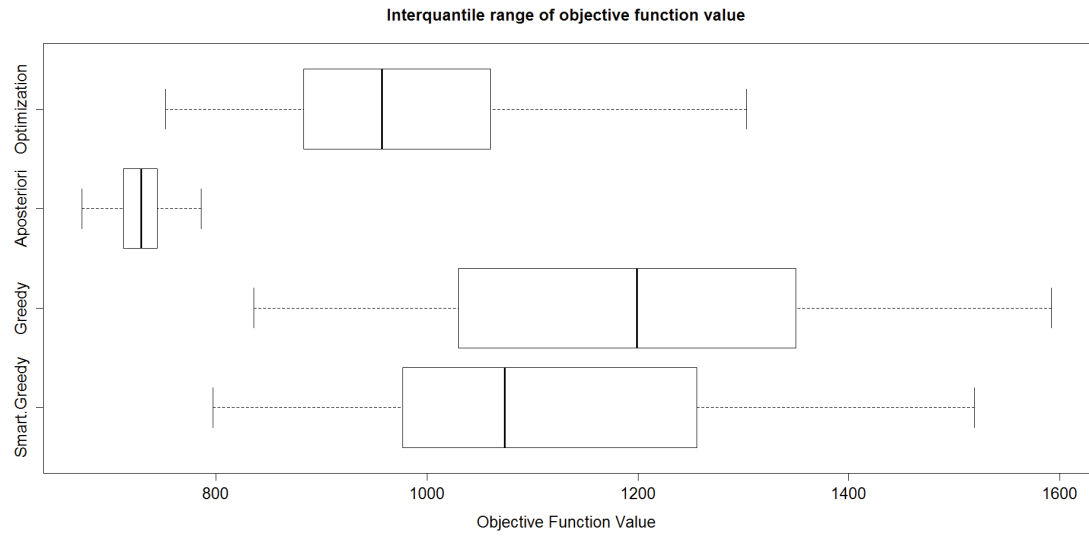


Figure 5.8: Comparison of the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between  $t=H$  and  $t=2H$  for the scenario with high congestion and low stochasticity

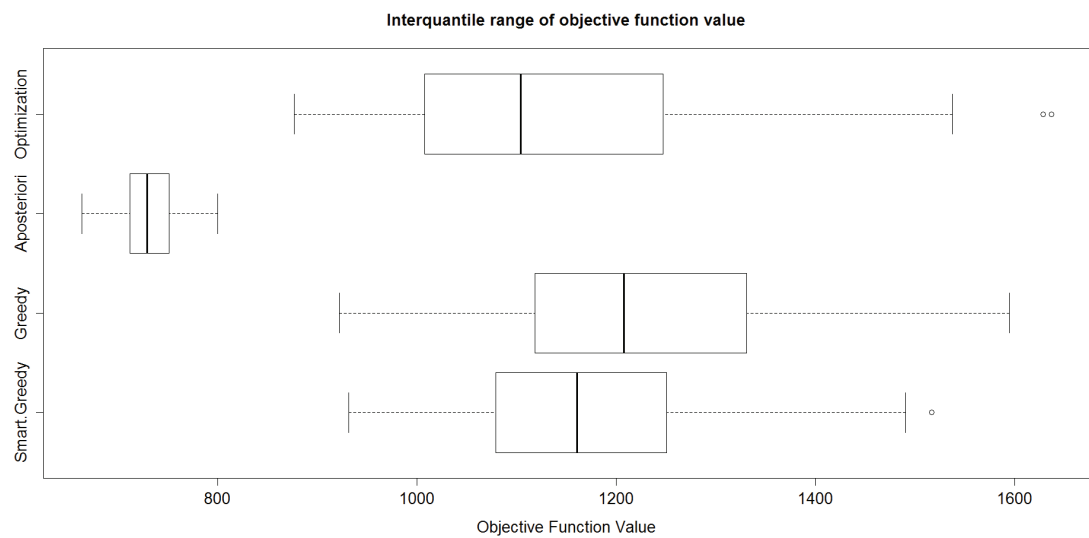


Figure 5.9: Comparison of the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between  $t=H$  and  $t=2H$  for the scenario with high congestion and high stochasticity

	Low Stochasticity			High Stochasticity		
	Optimization	Smart Greedy	Greedy	Optimization	Smart Greedy	Greedy
% difference in mean total cost	2.77%	14.81%	16.82%	3.75%	21.61%	25.23%
Number of unserved vessels at $t = H$	0	0	0	0	0	0
Number of unserved vessels at $t = 2H$	0	0	0	0	0	0
Average waiting time per vessel	0.12	1.43	1.67	0.32	2.72	3.03
Average total cost of deviation	33.96	24.77	41.99	60.404	42.78	57.42

Table 5.5: Performance indicators for the mildly congested scenario from  $t=H$  to  $t=2H$  averaged over 100 simulations runs

	Low Stochasticity			High Stochasticity		
	Optimization	Smart Greedy	Greedy	Optimization	Smart Greedy	Greedy
% difference in mean total cost	33.97%	51.40%	64.45%	55.80%	61.07%	67.29%
Number of unserved vessels at $t = H$	1.46	1.58	1.83	1.91	2.49	2.28
Number of unserved vessels at $t = 2H$	1.75	2.22	2.49	2.93	3.03	3.09
Average waiting time per vessel	17.10	19.67	21.45	23.02	23.40	24.62
Average total cost of deviation	281.66	236.3	303.52	549.64	357.83	405.48

Table 5.6: Performance indicators for the highly congested scenario from  $t=H$  to  $t=2H$  averaged over 100 simulations runs

### Including Cost of Deviation from the Original Schedule

In Figures 5.10-5.13, the results are shown when the cost term related to the deviation of the schedule is explicitly considered in the objective function. It can be observed that the proposed optimization based and smart greedy recovery methods significantly outperform the traditional greedy method of reassigning vessels at the port. It can be seen that in general, the solution performance of all the recovery methods deteriorates with increase in the level of congestion and the degree of stochastic variability. The optimization based recovery method outperforms the other recovery methods in all scenarios except the one with high congestion and high stochasticity. Thus it may be inferred that the optimization based recovery method is most sensitive to increase in the level of congestion and degree of stochasticity.

For the mildly congested scenarios, the optimization based recovery method was found to outperform the other two recovery methods in 96% and 91% of the simulation runs for the low and high stochasticity scenarios respectively. In the highly congested case on the other hand, the optimization based method returns the best solution in 57% and 28% of the simulation runs for the low and high stochasticity scenarios respectively. As can be seen from the Tables 5.7-5.8, it also does better in terms of the average waiting time and the number of unserved vessels in all except one scenario. In terms of adhering to the original schedule, the superiority of the smart greedy method is established, which performs the best in all but one scenario.

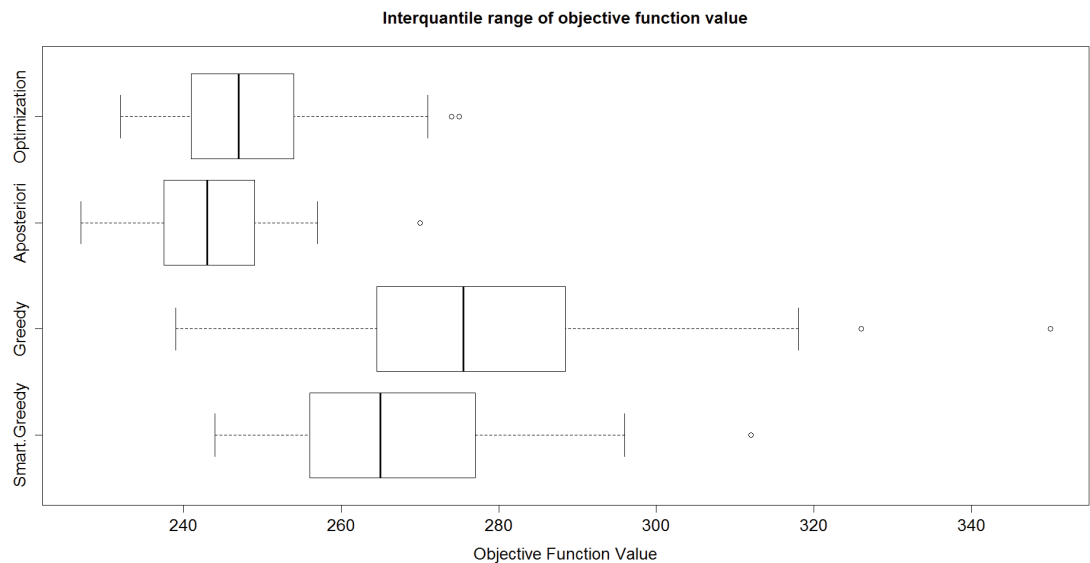


Figure 5.10: Comparison of the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between  $t=H$  and  $t=2H$  for the scenario with mild congestion and low stochasticity

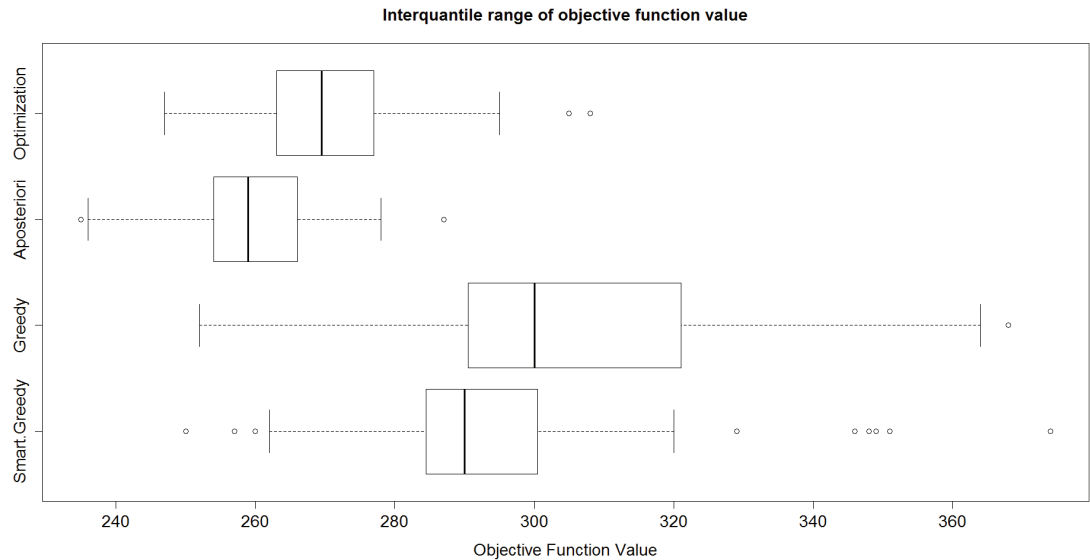


Figure 5.11: Comparison of the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between  $t=H$  and  $t=2H$  for the scenario with mild congestion and high stochasticity

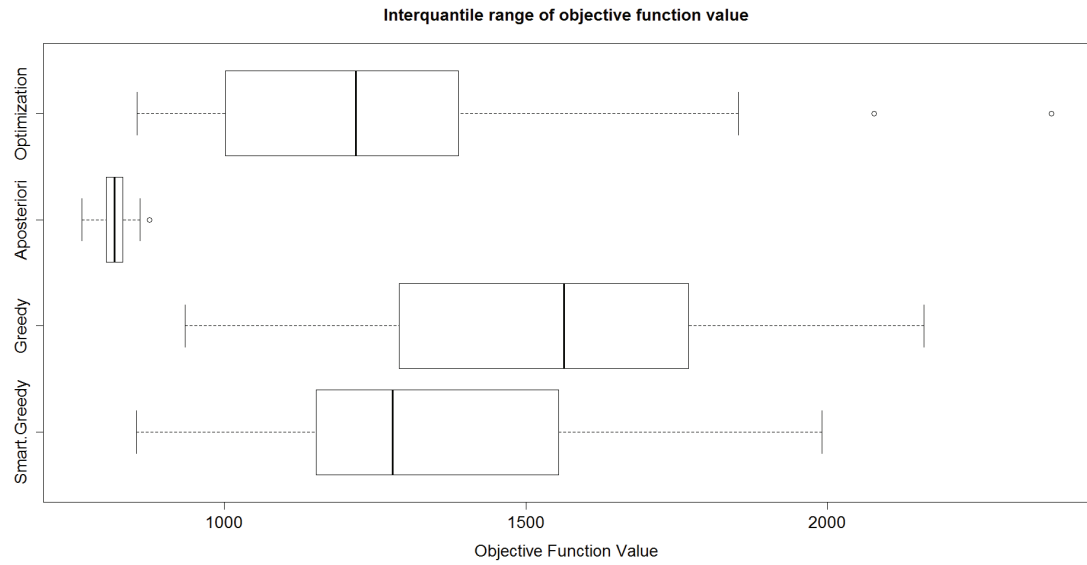


Figure 5.12: Comparison of the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between  $t=H$  and  $t=2H$  for the scenario with high congestion and low stochasticity

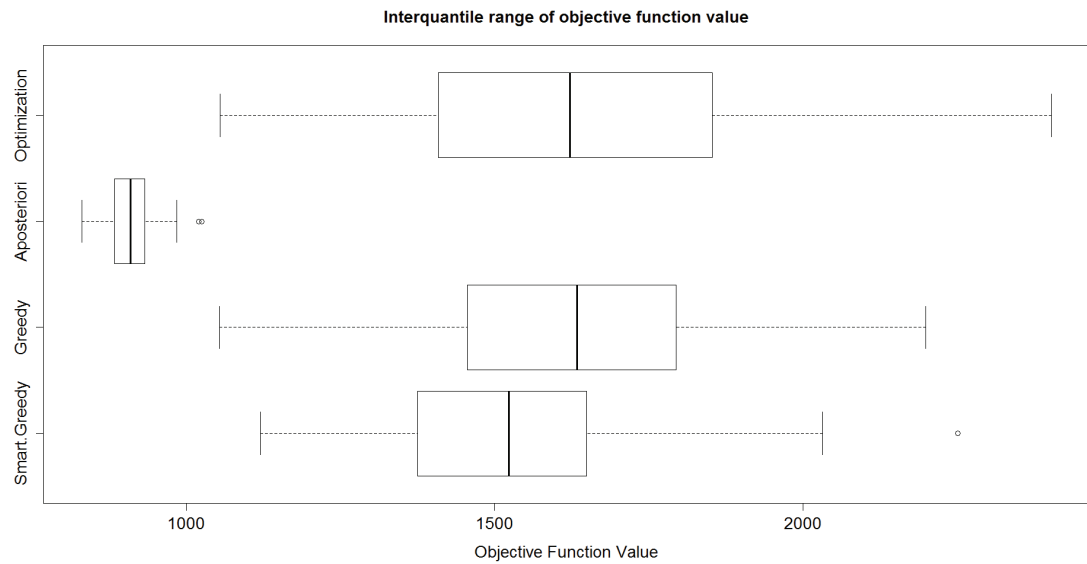


Figure 5.13: Comparison of the compare the solution performance of the algorithms based on 100 simulation runs for the vessels scheduled to arrive between  $t=H$  and  $t=2H$  for the scenario with high congestion and high stochasticity

	Low Stochasticity			High Stochasticity		
	Optimization	Smart Greedy	Greedy	Optimization	Smart Greedy	Greedy
% difference in mean total cost	1.78%	9.53%	13.75%	4.11%	13.27%	17.70%
Number of unserved vessels at $t = H$	0	0	0	0	0	0
Number of unserved vessels at $t = 2H$	0	0	0	0	0	0
Average waiting time per vessel	0.283	1.426	1.5	0.636	2.681	2.863
Average total cost of deviation	28.87	24.38	34.97	50.79	42.3	49.3

Table 5.7: Performance indicators for the mildly congested scenario from  $t=H$  to  $t=2H$  averaged over 100 simulations runs

	Low Stochasticity			High Stochasticity		
	Optimization	Smart Greedy	Greedy	Optimization	Smart Greedy	Greedy
% difference in mean total cost	48.06%	63.68%	86.85%	78.41%	68.88%	77.57%
Number of unserved vessels at $t = H$	1.6	1.58	1.93	2.26	2.49	2.10
Number of unserved vessels at $t = 2H$	1.89	2.21	2.55	3.09	2.97	3.08
Average waiting time per vessel	17.85	19.65	21.82	23.60	23.34	24.38
Average total cost of deviation	209.1	234.74	313.11	459.25	355.53	392.88

Table 5.8: Performance indicators for the highly congested scenario from  $t=H$  to  $t=2H$  averaged over 100 simulations runs



## 5.6 Conclusions and Future Work

In this work, we study and solve the problem of recovering a baseline berthing schedule of vessels at a port in real time as disruptions occur. To the best of our knowledge, there are very few studies that address the problem of real time recovery in port operations, which is typically based on local rescheduling heuristics or simple rules of thumb.

In our study, the underlying model is the dynamic hybrid berth allocation model developed in the context of bulk ports. The uncertainty in the unknown arrival times and handling times of the vessels is modeled based on probability distributions derived from past data. We present an optimization based recovery algorithm based on set partitioning and a heuristic based smart greedy recovery method to reschedule the vessels on a rolling time horizon for a given baseline schedule. The solution performance of the algorithms is tested and validated by conducting a simulation study in which the baseline schedule is the solution of the deterministic berth allocation problem. The results suggest that the proposed methodology for modeling the uncertainty, and the recovery algorithms can significantly reduce the total realized costs of berthing the vessels in comparison to the ongoing practice of re-assigning vessels at the port. The results further indicate that the optimization based method outperforms the other recovery methods in terms of the objective function cost and in terms of some key indicators such as the number of unserved vessels and the average waiting time. However, in terms of the adherence to the originally planned schedule, the smart greedy method is the superior method.

In the future, more work needs to be done to come up with appropriate pricing strategies that can enable the port to earn revenue from the late arriving vessels. Another natural extension of the work done so far is to develop a robust formulation for the berth allocation problem with a certain degree of anticipation of delays and variability in information. The recovery algorithms developed in this research can be applied on both the deterministic and robust formulations and the solution performance can be compared to assess the added benefit of robustness.



## 6 The Robust Single Machine Scheduling Problem

In this chapter, we demonstrate the complexity in handling uncertainty in a proactive manner for the most basic scheduling problem, that is, the single machine scheduling problem. In the context of port operations planning, the single machine scheduling problem is analogous to the discrete berth allocation problem with a single berth that can handle at most one vessel at a given time. Our primary goal in this research is to demonstrate the challenge of building robustness into scheduling solutions, while keeping the problem simple enough to permit useful analysis.

### 6.1 Introduction

Scheduling involves the optimal allocation of scarce resources to activities over time. Scheduling problems are an integral part of planning in areas such as production, service, manufacturing and transportation. In the past few decades, the practical importance and complexity of the general scheduling problem has motivated a significant volume of research in a wide variety of scheduling environments, including production and manufacturing systems, and transportation and logistics systems. Using standard notation, scheduling problems include a set of  $n$  jobs that must be scheduled on a set of  $m$  machines subject to certain constraints to optimize a desired objective function. In reality one or more characteristics of the jobs may be uncertain due to factors such as worker performance variability, changes in the work environment, variability in tool quality, and a variety of other factors. In this chapter we study the most common configuration, i.e. the single machine scheduling using  $m = 1$ , with a particular focus on generating “robust” schedules.

The major emphasis in past scheduling research has been on deterministic problems in which the schedule is computed and fixed in advance assuming perfect knowledge of job-specific attributes such as release times, processing times and/or due dates. However, a major drawback of precomputed schedules is that even small deviations in job parameter values can disrupt the schedule and lead to significant system performance degradation. Thus it is desirable to generate schedules that are “robust” given task parameter uncertainty. Consider a schedule that is created off-line and then placed into operation. During its execution, a disturbance may render the planned schedule infeasible. In response to the disturbance, a control action is executed to restore feasibility. A robust schedule is an *a priori* schedule which maintains high system performance in the presence of stochastic disturbances given a policy for control actions. In this study, we use a simple control policy that shifts the disrupted schedule in time without altering the original planned sequence of jobs, which is particularly useful in situations where changing the sequence may result in additional cost.

In classical stochastic scheduling, uncertain job attributes are modeled as independent random variables with known distributions. The performance of a schedule is dependent on the specific realization of each uncertain parameter during execution, while the design objective typically is to optimize the expected performance of the system. There are drawbacks of this approach. First, it assumes knowledge of probability distributions for the uncertain parameters, which are often unknown and almost never precisely known and may be difficult to estimate. Moreover, the decision maker may be more interested in hedging against the worst-case performance of the system than optimizing the system performance averaged over all possible realizations. However classical stochastic programming approaches fail to recognize this fact.

In this work, we study robust scheduling to determine a schedule which has the best worst-case performance. Our focus is single machine scheduling where the performance criterion is the total flow time of all jobs. The rest of the chapter is organized as follows: Section 6.2 provides a brief literature review on the general scheduling problem with a particular focus on research work done in dealing with uncertainty in the context of machine scheduling problems. In Section 6.3, we formally define the framework of the robust single machine scheduling problem and provide some important insights into the deterministic and stochastic variants of the problem. In Section 6.4 we propose solution algorithms to obtain good solutions for the robust single machine scheduling problem with release times. In Section 6.5, we present computational results based on artificial instances to test and validate the efficiency of the proposed

algorithms and compare their solution performance from a computational perspective. Finally we give some concluding remarks in Section 7.1.

## 6.2 Literature Review

Comprehensive literature surveys on the general scheduling problem in a wide variety of scheduling environments can be found in Lawler (1976), Graham et al. (1979) and Blazewicz (1987). Graham et al. (1979) established a three-field notation  $\alpha|\pi|\gamma$  to simplify the categorization of different types of machine scheduling problems. In this notation, the parameters  $\alpha$ ,  $\pi$  and  $\gamma$  describe the machine environment, the job characteristics and the optimality criterion respectively. For example,  $1|r_j|\sum C_j$  denotes the variant of the problem in which there is a single machine, each job  $j$  is available for processing only at the release time  $r_j$  or later, and the objective is to minimize the sum of completion times of all jobs as given by  $\sum C_j$ . As another example,  $1|r_j, \text{prec}|\sum_j C_j - r_j$  denotes the problem of scheduling the jobs with precedence constraints and release times on a single machine with the objective to minimize the total flow times of all jobs. As it will be impossible to enumerate all the variants of the problem and out of the scope of this study, we refer to Graham et al. (1979) for a survey on the different types of scheduling problems in literature.

Research has addressed machine scheduling problems in which one or more aspects of the jobs such as release times, processing times and other job-related properties are random, or the machines are subject to random breakdowns, or both. Glazebrook (1979), Weiss and Pinedo (1980), Emmons and Pinedo (1990) are few examples of such works. Stochastic machine scheduling problems focusing on probabilistic times have been studied by Wu and Zhou (2008), Skutella and Uetz (2005), Cai and Zhou (2005) and Soroush and Fredendall (1994) in which the job attributes are modeled as independent random variables with given distributions, whose actual values are realized during the execution of the schedule after a scheduling decision has been made. Dynamic scheduling methods in which jobs are dispatched dynamically to account for random disruptions in real time are studied by Gittins and Glazebrook (1977), Pinedo (1983), Glazebrook (1981), Glazebrook (1985) and few others. Another line of research focuses on responding to random disruptions that occur in real time, making it impossible to adhere to the originally planned schedule. Bean et al. (1987) and Roundy et al. (1989) are examples of such works. For detailed literature surveys on fundamental approaches for scheduling under uncertainty, refer to Herroelen and Leus (2005), Mohring et al. (1985), Mohring et al. (1984)

and Pinedo and Schrage (1982).

Kouvelis and Yu (1997) developed robust versions of many traditional discrete optimization problems. In general three different measures of robustness can be defined; one that minimizes the maximum absolute cost over the set of possible outcomes, a second that minimizes the maximum regret, i.e. the absolute difference in the solution cost between the realized outcome and the corresponding optimal solution for the outcome, and a third that minimizes the maximum relative deviation of the realized outcome from the corresponding optimal solution. Daniels and Kouvelis (1995) study the robust single machine scheduling problem without release times in which schedule robustness is measured by the absolute or relative deviation of the realized cost from optimality. They describe properties of robust schedules which allow the selection of a finite set of scenarios from uncertainty intervals of processing times to determine the worst-case deviation from optimality for a given schedule, and propose exact and heuristic solution approaches to obtain robust schedules. Yang and Yu (2002) study the same problem as Daniels and Kouvelis (1995), show that the problem is NP-hard even in the case of two scenarios for all three measures of robustness described earlier, and propose two alternative heuristic methods to obtain robust schedules. Kasperski (2005) studies the single machine scheduling problem for the absolute deviation measure of robustness, the maximum lateness performance criterion, and uncertainty intervals for the processing times. A polynomial time algorithm is proposed to solve the problem. More recently, Lu et al. (2012) study the single machine scheduling problem with uncertainty in the job processing times and sequence-dependent family setup times. In their study, the performance criterion is the total flow time of jobs, and the measure of schedule robustness is the maximum absolute deviation from the optimal solution in the worst-case scenario. They reformulate the problem as a robust constrained shortest path problem and propose a simulated annealing-based algorithm to determine robust schedules.

In this research, we use the maximum absolute cost over the set of all possible outcomes as the measure of robustness and the total flow time of jobs as the performance criterion to create robust schedules in the context of the single machine scheduling problem. To the best of our knowledge, this is the first study that considers uncertainty in both release times and processing times in the context of the robust single machine scheduling problem. We discuss some relevant properties of robust schedules with zero and non-zero release times, demonstrate the added complexity when release times are considered, propose an exact method to instantaneously solve the deterministic variant of the single

machine scheduling problem with release times, and develop heuristic methods based on variable neighborhood search and iterated local search to generate robust schedules. The solution performance of the proposed algorithms are tested and validated through extensive numerical experiments based on artificial data.

## 6.3 Robust Single Machine Scheduling Problem

### 6.3.1 Problem Definition

We consider a set of  $n$  jobs that are required to be scheduled on a single machine. We are interested in generating robust schedules for uncertain scheduling environments, in which there is stochastic variability in the release times  $r_i$  and the processing times  $p_i$  of jobs. In our problem, the release times and the processing times of the jobs are specified as independent ranges of values with unknown probability distributions, such that the release time interval of job  $i$  is  $[\underline{r}_i, \bar{r}_i]$  and the processing time interval of job  $i$  is  $[\underline{p}_i, \bar{p}_i]$ . Let the infinite set of possible realizations of release times and processing times be represented by the set  $\Omega$ . Then a possible outcome  $\lambda \in \Omega$ , represents a unique set of release times and processing times of the jobs, that can be realized with a certain positive and unknown probability. Let the decision space consisting of all possible job sequences be given by the set  $P$ . The cost of making sequencing decision  $\pi \in P$  under scenario  $\lambda \in \Omega$  is given by  $f(\pi, \lambda)$ . The optimal decision and the optimal cost under scenario  $\lambda \in \Omega$  are given by  $\pi_{\lambda^*}$  and  $f^*(\lambda)$  respectively.

We assume the following input data to be available for the single machine scheduling problem :

- $N =$  set of jobs
- $i = 1, \dots, |N|$  jobs
- $\Omega =$  the infinite set of possible realizations
- $P =$  decision space representing the set of all possible sequences
- $r_i^\lambda =$  release time of job  $i \in N$  for the realization  $\lambda \in \Omega$
- $p_i^\lambda =$  processing time of job  $i \in N$  for the realization  $\lambda \in \Omega$

The objective in the absolute robust single machine scheduling problem (AR-SMSP), can be mathematically expressed as follows

$$(\text{ARSMSP}) \min_{\pi \in P} \{ \max_{\lambda \in \Omega} (f(\pi, \lambda)) \} \quad (6.1)$$

## Chapter 6. The Robust Single Machine Scheduling Problem

The only decision variables in the above problems are the starting times of processing of jobs, as given by  $s_i$  for job  $i \in N$ . Let  $N_\pi$  represent the ordered sequence of jobs for the sequence  $\pi \in P$ , such that for jobs  $i, j \in N_\pi$  and  $j > i$ , it is implied that job  $j$  is sequenced after job  $i$  in  $\pi$ . For a given sequence  $\pi \in P$ , realization  $\lambda \in \Omega$  and the performance criterion as the total flow time of jobs, we have

$$f(\pi, \lambda) = \sum_{i \in N_\pi} (s_i - r_i^\lambda + p_i^\lambda) \quad (6.2)$$

subject to the conditions

$$s_1 = r_1^\lambda \quad (6.3)$$

$$s_i = \max(r_i^\lambda, s_{i-1} + p_{i-1}^\lambda) \quad \forall i \in N_\pi, i \geq 2 \quad (6.4)$$

The deterministic single machine scheduling problem (DSMSP) to determine  $f^*(\lambda)$  for a given realization  $\lambda \in \Omega$  can be formulated as follows:

$$(\text{DSMSP}) \quad \min \sum_{i \in N} (s_i - r_i^\lambda + p_i^\lambda) \quad (6.5)$$

$$\text{s.t. } s_i - r_i^\lambda \geq 0 \quad \forall i \in N \quad (6.6)$$

$$s_j \geq s_i + p_i^\lambda \parallel s_i \geq s_j + p_j^\lambda \quad \forall i, j \in N, i \neq j \quad (6.7)$$

In the above formulation, constraints (6.6) ensure that the processing of a job starts only at or after the release time of the job. Constraints (6.7) are the disjunctive constraints that ensure that two jobs are not processed at the same time. Unfortunately the disjunctive constraints are non-linear, but can be linearized using the big-M approach, and reformulated as

$$s_j + M(1 - z_{ij}) \geq s_i + p_i^\lambda \quad \forall i, j \in N, i \neq j \quad (6.8)$$

$$z_{ij} + z_{ji} = 1 \quad \forall i, j \in N, i \neq j \quad (6.9)$$

where  $z_{ij}$  is a binary variable equal to 1 if job  $i$  precedes job  $j$  without overlapping, 0 otherwise, and  $M$  is a large positive constant. With regard to complexity, DSMSP is strongly NP-hard (Lenstra et al. (1977)).



In the following section, our aim is to discuss some of the most important results related to the deterministic and robust variants of the single machine scheduling problem, and demonstrate the added complexity when there is uncertainty in both the release times and the processing times of the jobs. We begin by briefly looking at the deterministic version of the single machine scheduling problem without release times.

#### 6.3.2 Scheduling without release times

##### Deterministic Problem

The simplest scheduling problem arises when the release times of all jobs are equal to zero. The obvious approach to solve this problem is to assign a priority to each job based on the optimality criterion, and assign the jobs in the order of decreasing priorities whenever the machine becomes available. Note that in the absence of release times, the flow time of a given job is equivalent to its completion time. Thus according to the notation discussed earlier, the single machine scheduling problem without release times with the objective to minimize the total flow times can be represented by  $1 \mid C_j$ . Intuitively, it makes sense to schedule the job with the shortest processing time at the beginning so that the delays to all the other jobs are minimized, and in a similar way, schedule the remaining jobs in the order of increasing processing times. In the literature, this is commonly known as the *Shortest Processing Time* (SPT) rule. We have the following useful result (Smith (1956)).

RESULT 1: *SPT rule is an exact algorithm to solve  $1 \mid \sum C_j$  with time complexity  $O(n \log n)$ .*

##### Properties of robust schedules without release times

In the following discussion, we discuss some properties of robust schedules with the performance criterion as the total flow time or completion time (both are equivalent for zero release times) of the jobs. The release time of each job  $i \in N$  is equal to zero, and the processing time interval of job  $i$  is  $[p_i, \bar{p}_i]$ .

**ARSMSP with release times** We begin with a simple result for the absolute robust single machine scheduling problem (ARSMSP) without release times.

RESULT 2: *The optimal solution to the ARSMSP without release times is the sequence*

of jobs obtained by arranging the jobs in increasing order of  $\bar{p}_i$ , that is the highest processing time values for all jobs.

Proof: Let the sequence of jobs obtained by arranging the jobs in increasing order of the highest processing times be  $\pi_{\lambda_{\max}}$ . The worst case contingency for this sequence corresponds to the case when each job  $i \in N$  assumes its highest processing time  $\bar{p}_i$ . However it is obvious that the sequence  $\pi_{\lambda_{\max}}$  is also the optimal decision for the realization corresponding to this worst case contingency (using SPT algorithm discussed earlier). Hence for any other sequence  $\pi \in P$ , the flow time for the worst case contingency corresponding to  $p = \bar{p}_i$  for each job  $i \in N$ , is higher than for the sequence  $\pi_{\lambda_{\max}}$ . This proves the result.

### 6.3.3 Scheduling with release times

#### Deterministic Problem

As discussed earlier, the deterministic single machine scheduling problem (DSMSP) with release times is an NP-complete problem. Thus it may not be possible to obtain optimal solutions for large problems in a reasonable computation time by directly solving the MIP formulation of DSMSP as given by (6.5)-(6.7). In order to solve the robust single machine scheduling problem (RSMSP) with release times, it is desirable that we develop an efficient algorithm to solve DSMSP, that returns the optimal solution or at the very least a tight upper bound in a small computation time even for large problems. This point is further illustrated by the following result.

**RESULT 3:** *The maximum optimal value  $f^*(\lambda)$  over the set of all possible realizations  $\lambda \in \Omega$  is a lower bound to the absolute robust single machine scheduling problem (ARSMSP) with (or without) release times.*

Proof: Let's say that we are given a sequence  $\pi \in P$ , for which  $\lambda_\pi$  is the worst case realization. Then we have

$$f(\pi, \lambda_\pi) \geq f(\pi, \lambda) \quad \forall \lambda \in \Omega \quad (6.10)$$

Let  $f^*(\lambda)$  be the optimal value of the flow time for the realization  $\lambda \in \Omega$ . Then

by definition, we also have

$$f(\pi, \lambda) \geq f^*(\lambda) \quad \forall \lambda \in \Omega \quad (6.11)$$

Using 6.10 and 6.11 we have,

$$f(\pi, \lambda_\pi) \geq f^*(\lambda) \quad \forall \lambda \in \Omega \quad (6.12)$$

The above inequality implies that for any sequence  $\pi \in P$ , the flow time corresponding to the worst case realization is greater than or equal to the optimal flow times for all realizations  $\lambda \in \Omega$ . Since the above inequality holds for all  $\pi \in P$ , it can be equivalently written as

$$\min_{\pi \in P} f(\pi, \lambda_\pi) \geq \max_{\lambda \in \Omega} f^*(\lambda) \quad (6.13)$$

Note that the left hand side of the above inequality is the objective of the ASMRSP. This proves the result.

In past research, significant success has been achieved in developing approximation algorithms for  $1|r_j|\sum C_j$  i.e. DSMSP with release times to minimize the total completion time of jobs. The best known approximation algorithm for  $1|r_j|\sum C_j$  by Phillips et al. (1998) is a 2-approximation algorithm that produces non-preemptive schedules from optimal preemptive schedules which can be easily determined using the *Shortest Remaining Processing Time* (SRPT) rule. It may be noted that for a given vector of release times and processing times, the optimal solution for  $1|r_j|\sum C_j$  is also the optimal solution for  $1|r_j|\sum C_j - r_j$ . However the approximability of these two criteria may be very different as shown by Kellerer et al. (1999). Some of the reasonable approximation algorithms for  $1|r_j|\sum C_j - r_j$  are the *Earliest Start Time* (EST) rule in which the shortest available job is assigned whenever the machine becomes free for assignment, or the *Earliest Completion Time* (ECT) rule in which the job with the earliest completion time (that may not be available yet) is assigned to the machine. Both the rules have a worst-case performance bound of  $O(n)$ . Kellerer et al. (1999) proposed an approximation algorithm with a sub-linear worst-case

Job 1	1	1	1	0
Job 2	0	0	0	1
Time 1	1	0	0	0
Time 2	1	1	0	0
Time 3	0	1	1	1
Time 4	0	0	1	1

Table 6.1: Assignment matrix for a simple example of set partitioning problem

performance guarantee of  $O(n^{1/2})$ . They further showed that no constant ratio approximation algorithm can be expected for this problem by proving that there exists no polynomial time approximation algorithm with a worst-case performance bound of  $O(n^{1/2-\epsilon})$ , for any  $\epsilon \geq 0$ . It is clear that the bound obtained from the best known approximation algorithm is extremely weak for the problem under study. In the following section, we propose an exact method based on set-partitioning to solve the DSMSP with release times to optimality with a computation time that is instantaneous for even large problem size.

**Exact Algorithm based on Set Partitioning** As discussed earlier, *Result 3* necessitates an exact method to solve the deterministic single machine scheduling problem (DSMSP) with release times to get a lower bound on the ARSMSP with release times. In this section, we propose an exact method based on set-partitioning to solve large instances of the DSMSP with release times in small computation time. In this method, the set of all feasible assignments is generated apriori and is denoted by the set  $J$ . The assignment matrix is composed of the upper submatrix  $A$  and lower submatrix  $B$ . The upper submatrix  $A$  consists of  $|J|$  columns and  $|N|$  rows. In submatrix  $A$ , if column  $j \in J$  represents the feasible assignment of job  $i \in N$ , then the entry in row  $i$  is 1 while all other entries are zeroes. The lower submatrix  $B$  consists of  $|J|$  columns and a single row for every discrete time interval in the planning horizon. Thus, in submatrix  $B$ , if column  $j \in J$ , represents the feasible assignment of job  $i \in N$ , then all entries corresponding to the time intervals in which the job  $i$  is processed in the feasible assignment  $j \in J$  are 1, while all the remaining entries are zeroes. To illustrate the procedure for the specific problem we are solving, consider the example containing two jobs, and four discrete time intervals in the planning horizon. Let us assume that both jobs have processing times of two time units, job 1 is released at time 1, while job 2 is released at the start of time 3, and hence can only be processed after that. Then the assignment matrix for the problem would look like as shown in Table 6.1. The first column represents the assignment of job 1 from time 1-2, and so on.

### 6.3. Robust Single Machine Scheduling Problem

We assume the following input data to be available for the set partitioning model:

- $N =$  set of jobs
- $H =$  set of discrete time intervals in the planning horizon
- $J =$  set of feasible assignments
- $t = 1, \dots, |H|$  discrete time intervals in the planning horizon
- $j = 1, \dots, |J|$  feasible assignments
- $d_j =$  delay associated with assignment  $j$
- $h_j =$  processing time associated with assignment  $j$

The assignment matrix coefficients are defined as follows.

$$A_{ij} = \begin{cases} 1 & \text{if the feasible assignment } j \text{ represents job } i; \\ 0 & \text{otherwise.} \end{cases}$$

$$B_j^t = \begin{cases} 1 & \text{if job is being processed in time interval } t \text{ in assignment } j; \\ 0 & \text{otherwise.} \end{cases}$$

There is only a single decision variable for selection of feasible assignments in the optimal solution which is defined as follows.

$$\lambda_j = \begin{cases} 1 & \text{if assignment } j \text{ is part of the optimal solution;} \\ 0 & \text{otherwise.} \end{cases}$$

The set partitioning model to solve the single machine scheduling problem with release times is formulated as shown below:

$$\min \sum_j (d_j \lambda_j + h_j \lambda_j) \tag{6.14}$$

$$\text{s.t. } \sum_j (A_{ij} \lambda_j) = 1 \quad \forall i \in N \tag{6.15}$$

$$\sum_j (B_j^t \lambda_j) \leq 1 \quad \forall t \in H \tag{6.16}$$

$$\lambda_j \in \{0, 1\} \quad \forall j \in J \tag{6.17}$$

In the above model, the objective (6.14) is to minimize the total flow time of the jobs, which includes the delays and the total processing times of the jobs. Note that the objective function can be equivalently expressed as the minimization of the sum of delays only, since the sum of processing times of the jobs given

by  $\sum_j (h_j \lambda_j)$  is a constant. Thus in the proposed set partitioning model, the processing times are only used to build the matrix  $B$ . Constraints (6.15) ensure that each job must have exactly one feasible assignment in the optimal solution. Constraints (6.16) ensure that in a given time interval, at most one job can be processed. While the growth in the number of variables and constraints in the set-partitioning approach is much faster as compared to the mixed integer programming formulation discussed earlier, it can be used to obtain optimal solutions to the DSMSP almost instantaneously for even large problem size.

### Robust Scheduling with release times

In the following discussion, we discuss some properties of robust schedules with the performance criterion as the total flow time of the jobs. The release time of each job  $i \in N$  lies in the interval  $[\underline{r}_i, \bar{r}_i]$ , and the processing time interval of job  $i$  is  $[\underline{p}_i, \bar{p}_i]$ .

**ARSMSP with release times** The absolute robust single machine scheduling problem (ARSMSP) with release times can be mathematically formulated as follows:

$$(\text{ARSMSP}) \min_{\pi \in P} \{ \max_{\lambda \in \Omega} (f(\pi, \lambda)) \} \quad (6.18)$$

subject to the conditions

$$f(\pi, \lambda) = \sum_{i \in N_\pi} (s_i - r_i^\lambda + p_i^\lambda) \quad (6.19)$$

$$s_1 = r_1^\lambda \quad (6.20)$$

$$s_i = \max(r_i^\lambda, s_{i-1} + p_{i-1}^\lambda) \quad \forall i \in N_\pi, i \geq 2 \quad (6.21)$$

In order to determine the sequence with the best worst-case absolute performance, we first formulate the problem of evaluating the worst case scenario for a given sequence  $\pi \in P$ . Note that it is not straightforward to solve this problem by a simple enumeration technique, since the release times and processing times of all jobs are specified as independent ranges, thus implying an infinite number of possible realizations. However we have the following useful result that allows us to restrict our attention to only a subset of the realizations. The

### 6.3. Robust Single Machine Scheduling Problem

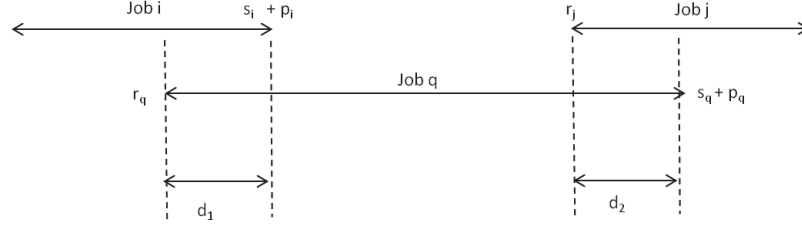


Figure 6.1: Jobs  $i$ ,  $q$  and  $j$  in an ordered sequence

result can be stated as follows

**RESULT 4:** For the ARSMSP with  $n$  jobs and uncertainty in both release times and processing times of jobs, there exists a worst-case scenario  $\lambda_\pi$  for sequence  $\pi \in P$ , that belongs to a subset of cardinality  $2^{n-2}$  of the extreme point scenarios of  $\pi$ .

**Proof:** Consider an ordered sequence  $N_\pi$  of jobs, in which jobs  $i$ ,  $q$  and  $j$  are consecutively ordered, that is,  $i \prec q \prec j$ . When job  $q$  is the first or the last job in the sequence, jobs  $i$  and  $j$  respectively, may be considered as fake jobs. We assume that the release times and processing times of all jobs in the sequence except job  $q$  are given (and unchangeable), and we want to show that there is an extreme point scenario of release time and processing time corresponding to job  $q$ , for which the job sequence assumes its worst case value. We define the following notations to illustrate the proof. Let  $d_1$  be the overlap between the release time of job  $q$  given by  $r_q$  and the finishing time of processing job  $i$  as given by  $s_i + p_i$ . Similarly let  $d_2$  be the overlap between the end time of processing job  $q$  given by  $s_q + p_q$  and the release time of job  $j$  given by  $r_j$ . This is graphically shown in Figure 6.1.

To obtain the worst case value, we need to maximize the sum of  $d_1$  and  $d_2$ . We consider the following three cases:

- Case I: Job  $q$  is the first job in the sequence. In this case,  $d_1 = 0$  and  $d_2 = \max(0, s_q + p_q - r_j)$ . It is easy to see that there is a worst case scenario corresponding to  $p_q = \overline{p}_q$  and  $r_q = \overline{r}_q$ .
- Case II: Job  $q$  is the last job in the sequence. In this case,  $d_1 = \max(0, s_i + p_i - r_q)$  and  $d_2 = 0$ . Again, it is easy to see that there is a worst case scenario corresponding to  $r_q = \underline{r}_q$  and  $p_q = \overline{p}_q$ .
- Case III: When job  $q$  lies somewhere in between,  $d_1 = \max(0, s_i + p_i - r_q)$  and  $d_2 = \max(0, s_q + p_q - r_j)$ . By inspection, it can be inferred that  $d_1 + d_2$

is maximized when  $p_q = \overline{p}_q$  and  $r_q = \overline{r}_q$  or  $\underline{r}_q$ .

Summarizing the above cases, there is a single unique endpoint scenario corresponding to the worst case contingency in cases I and II. For case III, for each of the  $n-2$  possible positions of job  $q$  in the sequence, there are 2 possible realizations of release times and a single realization of processing time for which the worst case value of the sequence may be obtained. Thus for  $n$  jobs in a given sequence, there exists a worst case scenario belonging to a subset of cardinality  $2^{n-2}$  of the set of all possible realizations. This proves the result.

The above result indicates that in order to determine the worst case scenario for a given sequence from the set of infinite possible realizations of release times and processing times, attention can be restricted to a subset of cardinality  $2^{n-2}$  of endpoint scenarios. However this number can also be significantly large for large value of  $n$ . In the following, we show that the problem of finding the worst case realization for a given sequence can be formulated and solved as a mixed integer linear program (MILP). The absolute worst case performance problem (AWCPP) for a given ordered sequence of jobs  $N_\pi$  can be stated as follows:

$$(AWCPP) \max \sum_{i \in N_\pi} (s_i - r_i^\lambda + p_i^\lambda) \quad (6.22)$$

$$s_1 = r_1^\lambda \quad (6.23)$$

$$s_i = \max(r_i^\lambda, s_{i-1} + p_{i-1}^\lambda) \quad \forall i \in N_\pi, i \geq 2 \quad (6.24)$$

$$r_i^\lambda \in [\underline{r}_i, \overline{r}_i] \quad \forall i \in N_\pi \quad (6.25)$$

$$p_i^\lambda \in [\underline{p}_i, \overline{p}_i] \quad \forall i \in N_\pi \quad (6.26)$$

In the above model, constraints (6.23) state that the processing of the first job in the sequence starts as soon as it is released. The constraints (6.24) state that the processing of each subsequent job in the sequence should start as soon as the job is released and the processing of the previous job in the sequence has finished. The constraints (6.24) are not linear, but can be linearized using standard techniques (see Watters (1967)). To begin with we introduce two sets of additional variables  $\sigma_i$  and  $\gamma_i$  for all jobs  $i \in N$ . Then the constraints (6.24) can be equivalently expressed as

$$s_i = r_i^\lambda + \sigma_i \quad \forall i \in N_\pi, i \geq 2 \quad (6.27)$$

$$s_i = s_{i-1} + p_{i-1}^\lambda + \gamma_i \quad \forall i \in N_\pi, i \geq 2 \quad (6.28)$$



### 6.3. Robust Single Machine Scheduling Problem

$$\sigma_i \gamma_i = 0 \quad \forall i \in N_\pi, i \geq 2 \quad (6.29)$$

To linearize constraints (6.29) we introduce binary variables  $u_{ik}$  and  $v_{ik}$  for all jobs  $i \in N$ , for a large enough positive integer  $K$  such that  $k \leq K$ . Note that the product  $\sigma_i \gamma_i$  is of the form  $\sum_{t \leq K^2} \sum_{k \leq K} \sum_{j \leq K} C_t u_{ik} v_{ij}$ , where the  $C_t$  terms are constants. For the product  $\sigma_i \gamma_i$  to be equal to zero, each term  $C_t u_{ik} v_{ij}$  should be equal to zero. This entails one or both the binary variables,  $u_{ik}$  and  $v_{ij}$ , to be equal to zero. This can be mathematically modeled as  $u_{ik} + v_{ij} \leq 1$ . Thus we have the linearized version,

$$\sigma_i = \sum_{k \leq K} 2^k u_{ik} \quad \forall i \in N_\pi, i \geq 2 \quad (6.30)$$

$$\gamma_i = \sum_{k \leq K} 2^k v_{ik} \quad \forall i \in N_\pi, i \geq 2 \quad (6.31)$$

$$u_{ik} + v_{ij} \leq 1 \quad \forall i \in N_\pi, i \geq 2, \forall j, k \leq K \quad (6.32)$$

$$u_{ik}, v_{ik} \in \{0, 1\} \quad \forall i \in N_\pi, i \geq 2, \forall k \leq K \quad (6.33)$$

Following the above discussion, replacing  $\sigma_i$  and  $\gamma_i$  from constraints (6.30)-(6.31), the AWCPP can be rewritten as a mixed integer linear program as follows

$$(AWCPP) \max \sum_{i \in N_\pi} (s_i - r_i^\lambda + p_i^\lambda) \quad (6.34)$$

$$s_1 = r_1^\lambda \quad (6.35)$$

$$s_i = r_i^\lambda + \sum_{k \leq K} 2^k u_{ik} \quad \forall i \in N_\pi, i \geq 2 \quad (6.36)$$

$$s_i = s_{i-1} + p_{i-1}^\lambda + \sum_{k \leq K} 2^k v_{ik} \quad \forall i \in N_\pi, i \geq 2 \quad (6.37)$$

$$u_{ik} + v_{ij} \leq 1 \quad \forall i \in N_\pi, i \geq 2, \forall j, k \leq K \quad (6.38)$$

$$u_{ik}, v_{ik} \in \{0, 1\} \quad \forall i \in N_\pi, i \geq 2, \forall k \leq K \quad (6.39)$$

$$r_i^\lambda \in [\underline{r}_i, \bar{r}_i] \quad \forall i \in N_\pi \quad (6.40)$$

$$p_i^\lambda \in [\underline{p}_i, \bar{p}_i] \quad \forall i \in N_\pi \quad (6.41)$$

Thus given a sequence  $\pi \in P$ , the worst case sequence can be determined by

## Chapter 6. The Robust Single Machine Scheduling Problem

---

solving the above MILP. Note that in the above formulation, for  $|N|$  jobs, the number of variables is of the order of  $|N||K|$  and the number of constraints is of the order of  $|N||K|^2$ . From the computational experiments, the above MILP was found to be solvable almost instantaneously for even large problem size. The ARSMSP with release times given by (6.18)-(6.21) is solved using heuristic techniques described in the following section.

## 6.4 Solution Algorithms to the ARSMSP with Release Times

In this section, we present two alternative heuristic methods to obtain optimal or near-optimal solutions for the absolute robust single machine scheduling problem with uncertainty in release times and processing times.

### 6.4.1 Iterated Local Search

To begin with, we implement a simple heuristic based on iterated local search. In this method, we start with a random initial solution and perform a local search on the neighborhood of this sequence. In our implementation, the local search neighborhood  $N_{LS}$  of a given sequence is defined as the set of sequences obtained by swapping two adjacent jobs in the original sequence. In case the local search improves the current solution, the local search solution is accepted as the new current solution and the local search is performed again. When the algorithm is stuck at a local minimum for too long, the algorithm is restarted with a new initial solution. The algorithm is terminated when the elapsed time from the beginning crosses a threshold computational time limit. The algorithm is described in Algorithm 10:

---

**Algorithm 10:** Iterated Local Search Algorithm

---

**Require:** Set  $N$  of jobs, set  $M$  of scenarios

Construct an initial feasible solution

currentBestSolution  $\leftarrow$  initialSolution

bestWorstCaseScenarioValue  $\leftarrow$

worstCaseScenarioValue(currentBestSolution)

**while** timeLimit  $\leq$  ilsTimeLimit **do**

$x' = \text{LocalSearch}(\text{currentBestSolution}, N_{LS})$

**if** worstCaseScenarioValue( $x'$ ) < bestWorstCaseScenarioValue **then**

        bestWorstCaseScenarioValue  $\leftarrow$  worstCaseScenarioValue( $x'$ )

        currentBestSolution  $\leftarrow x'$

**end if**

**if** solution value does not improve over time = timeRandomRestartILS

**then**

        reinitialize currentSolution and start all over

**end if**

**end while**

---

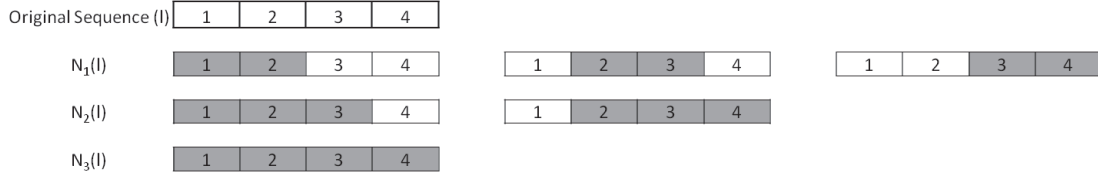


Figure 6.2: VNS Neighborhood Structures for a given sequence 1-2-3-4

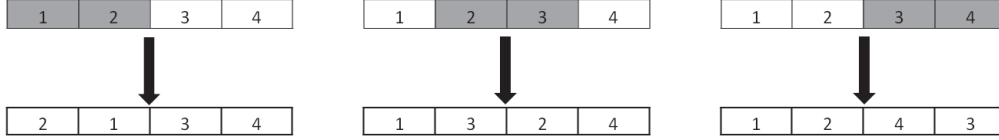


Figure 6.3: VNS Neighborhood  $N_1(1-2-3-4)$

### 6.4.2 Variable Neighborhood Search Algorithm

In this section, we propose the metaheuristic popularly known as the variable neighborhood search (VNS) in the literature. The algorithm was initially developed by Hansen and Mladenovic (1997). The main idea of the variable neighborhood search algorithm is to explore multiple neighborhood structures systematically instead of a single neighborhood, and escape local minima (in the case of minimization). In our implementation of the method, the  $k^{\text{th}}$  neighborhood structure,  $N_k(\ell)$  of a given sequence  $\ell$  is the set of sequences obtained by permuting the subset of jobs that are at most  $k$  indices apart in the original sequence. It naturally follows that a sequence containing  $n$  jobs has  $n-1$  neighborhood structures. This is graphically represented in the Figure 6.2, where the permutable subset of jobs are shown in the blocks shaded in grey. Note that the neighborhood structure  $N_1(\ell)$  contains three candidate solutions as shown in Figure 6.3

In the implementation of the VNS, we start with an initial feasible solution  $x$ . Iteratively starting from  $k=1$ , the *shaking procedure* is applied in which a random neighbor  $x'$  is generated in the  $N_k$  neighborhood of  $x$ . The shaking procedure is important as it prevents the algorithm from getting trapped at a local minimum. Thereafter a *local search* is carried out in the  $N_{LS}$  neighborhood of  $x'$ , where the  $N_{LS}$  neighborhood has a similar definition to the one described previously for the iterated local search method. If the local search solution  $x''$  is found to be better than the current solution  $x$ , the search continues with the local search solution  $x''$  as the new starting point, and  $k$  is re-initialized to be equal to 1. If no improvement is found in the  $N_k$  neighborhood, then  $x$  remains the start-

---

#### 6.4. Solution Algorithms to the ARSMSP with Release Times

---

ing point for randomly generating a neighboring solution from the subsequent neighborhood  $N_{k+1}$ . When the current solution does not improve over a certain predefined time limit, the whole procedure is repeated starting from  $k=1$  with a different initial solution. The algorithm is terminated when the time elapsed from the beginning crosses a threshold computational time limit. The algorithm is described in Algorithm 11:

---

**Algorithm 11:** Variable Neighborhood Search Algorithm

---

**Require:** Set  $N$  of jobs, set  $M$  of scenarios

Construct an initial feasible solution

currentBestSolution  $\leftarrow$  initialSolution

bestWorstCaseScenarioValue  $\leftarrow$

worstCaseScenarioValue(currentBestSolution)

**while** timeLimit  $\leq$  vnsTimeLimit **do**

$k=1$

**while**  $k \leq (|N|-1)$  **do**

*Shaking Procedure*

$x' = \text{GenerateNeighbor}(\text{currentBestSolution}, N_k)$

*Local Search*

$x'' = \text{LocalSearch}(x', N_{LS})$

**if** worstCaseScenarioValue( $x''$ )  $<$  bestWorstCaseScenarioValue **then**

      bestWorstCaseScenarioValue  $\leftarrow$  worstCaseScenarioValue( $x''$ )

      currentBestSolution  $\leftarrow x''$

$k=1$

**else**

$k++$

**end if**

**if** solution value does not improve over time = timeRandomRestartVNS

**then**

      reinitialize currentSolution and start all over

**end if**

**end while**

**end while**

---

Before proceeding to the computational results, we look at how depending on the problem size, a benchmark solution is obtained to assess the solution performance of the proposed heuristic techniques.

### 6.4.3 Methods to Determine the Benchmark Solution

In this section, we discuss methods to obtain optimal solutions for small instances of the ARSMSP with release times and lower bounds for large instances of the problem, to assess and compare the solution performance of the proposed heuristic algorithms.

#### Exhaustive Search Algorithm

In the exhaustive search method, all the possible  $n!$  sequences for  $n$  jobs are explicitly enumerated and the objective value is calculated for each sequence. Using *Result 4* discussed earlier in the chapter, for  $n$  jobs, the  $2^n$  extreme point scenarios corresponding to the two extreme point values of release times and the highest processing time value of each job need to be considered. Evidently as the problem size grows, it is computationally extremely expensive to use this method. However, it is useful to validate the solution performance of heuristics on small sized instances, for which optimal solutions can be obtained. The implementation of the method is described in Algorithm 12.

---

#### Algorithm 12: Exhaustive Search Algorithm

---

**Require:** Set  $N$  of jobs, set  $M$  of scenarios  
sequences = GenerateAllPossibleSequences( $N$ )  
**for** list  $i$  : sequences **do**  
    Calculate worstCaseScenarioValue( $i, M$ )  
**end for**  
Determine sequence with best worstCaseScenarioValue

---

#### Calculation of Lower Bound

As shown in *Result 3* earlier, the maximum optimal value over the set of all possible realizations is a lower bound to the absolute robust single machine scheduling problem (ARSMSP). From the computational experiments, it was found that for instances up to 15 jobs, the lower bound could be determined by brute force method in a reasonable computational time of about an hour. However for larger instances, the computational time may be very large. Thus in order to speed up the computation of the lower bound, we implement the following simple code similar to the iterated local search described earlier. We know that for each job, two extreme point values of the release times and a single value of the processing time need to be considered. Then for an ordered

## 6.4. Solution Algorithms to the ARSMSP with Release Times

sequence of jobs  $N_\pi$  containing jobs from 1 to  $n$ , a given scenario can be represented by a binary string, where a 0 represents the left side extreme value of the release time and value 1 represents the right side extreme value. This is graphically represented for the sequence 1-2-3-4 in Figure 6.4. For a given

$r_1 \in [5,10]$	$r_2 \in [4,8]$	$r_3 \in [6,9]$	$r_4 \in [7,12]$
1	2	3	4

$r_1 = 5$	$r_2 = 8$	$r_3 = 9$	$r_4 = 7$
0	1	1	0

$r_1 = 10$	$r_2 = 8$	$r_3 = 6$	$r_4 = 12$
1	1	0	1

Figure 6.4: Binary string representations for the job sequence 1-2-3-4, for two different extreme point scenarios

realization, the optimal solution value is computed using the set-partitioning method described earlier in the chapter. In the local search, a neighboring solution is obtained by switching a single job from 0 to 1, or vice versa. We perform a simple local search on a randomly chosen initial scenario, choose the solution with the highest optimal value in the neighborhood, which then becomes the new candidate scenario for local search and so on. When the algorithm is stuck at a local minimum, the whole procedure is restarted with a new randomized solution. The algorithm is terminated after a preset computational time and the best solution obtained thus far is accepted as the final solution. The algorithm was found to perform exceptionally well for the computation of the maximum optimal flow time, as indicated by the computational experiments on instances containing up to 15 jobs. In a computational time of less than a minute, the algorithm was found to return the exact value of the maximum optimal flow time as determined from the brute force method, while in a few instances there was a difference of less than 1 %.

## **6.5 Computational Results and Analysis**

### **6.5.1 Generation of Instances**

The proposed heuristic algorithms were tested and validated through extensive numerical experiments based on artificial instances. The algorithms were implemented in JAVA programming language, and computational tests were run on an Intel Core i7 (2.80 GHz) processor and used a 32-bit version of CPLEX 12.2.

The experimental design adopted for the computational study consists of test problems involving  $|N|=7, 15, 20, 30$  and  $50$  jobs and a single machine. For each problem size, 20 instances were tested. Based on the degree of stochastic variability in the release times and processing times of the jobs, the test instances are categorized into four different sets. For each category, the instances are generated by randomly drawing the lower and upper ends of the release time range and the processing time range of the jobs. The lower end of the release time range  $\underline{r}_i$  is drawn from a uniform distribution of integers on the interval  $\underline{r}_i \in [0, 5\beta]$  for four different values of  $\beta$  ( $\beta=2,3,4$  and  $6$ ). For  $\beta = 2$  and  $3$ ,  $\bar{r}_i$  is equal to  $\underline{r}_i + 10$ . On the other hand, for  $\beta = 4$  and  $6$ ,  $\bar{r}_i$  is equal to  $\underline{r}_i + 20$ . The lower end of the processing time range  $\underline{p}_i$  is drawn from a uniform distribution of integers on the interval  $[1,4]$ , while the upper end of the processing time range is equal to  $\underline{p}_i + 6$ . Five problem instances are tested for each combination of  $|N|$  and  $\beta$ , resulting in a total of 100 problem instances.

### **6.5.2 Discussion of Results**

The computational results obtained from the algorithms discussed previously are shown in the Tables 6.2-6.6. For  $|N|=7$  jobs, the optimal solution is calculated using an exhaustive search algorithm. Thus for test instances with  $|N|=7$  jobs, it is possible to determine the strength of the lower bound. As evident from Table 6.2, the lower bound is not too strong, and with increasing  $\beta$  value, implying a larger uncertainty in the release times of the jobs, the bound weakens. For large problem size, it can be expected that the bound is even weaker.

From the results tables, it can be inferred that in general, the variable neighborhood search (VNS) algorithm is the superior method to generate robust schedules. Based on a trial analysis, the computational time limit for test instances corresponding to a given combination of  $|N|$  and  $\beta$  was set to a certain value. It



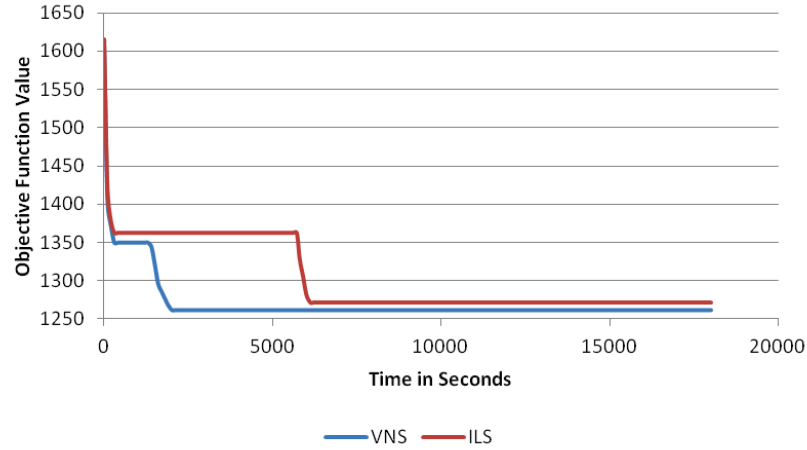


Figure 6.5: Convergence of Instance C18 over a computational time limit of 5 hours

can be seen that for  $|N|=7$ , the VNS algorithm is able to generate optimal solutions for all instances in a computational time of few seconds. The iterated local search (ILS) method on the other hand is able to generate optimal solutions for close to 50% of the problem instances in the computational time limit of 100 seconds. For larger problem size with  $|N| = 15, 20, 30$  and 50 jobs, the worst case value for a given sequence of jobs is determined by solving the mixed integer linear program (6.34)-(6.41) using  $K=10$ . On the average, the instances were found to converge faster for small  $\beta$  value, that is, smaller uncertainty in the release times of the jobs. As can be seen from the results, the VNS and ILS algorithms converge to approximately the same solution for a few instances. Although the gap with respect to the lower bound is pretty large for most test instances, but since the bound is a weak one as established previously, it is difficult to comment on the absolute solution performance of the algorithms for these instances. Figure 6.5 shows the convergence of the solution for the test instance C18 over a computational time limit of 5 hours for the VNS and ILS methods. Note that the solution value may remain stable for a long time, before it begins to improve again.

From the computational experiments it was found that there is a certain degree of variance in the output solution values when a given problem instance was tested using a given algorithm. To study the behavior of the algorithms in more depth, we conduct a simulation study in which a test instance is run 50 times using a given algorithm and the resulting output solution values are plotted against the associated probability of finding a solution in the corresponding output range of values. The plots for some of the instances are shown in Figures

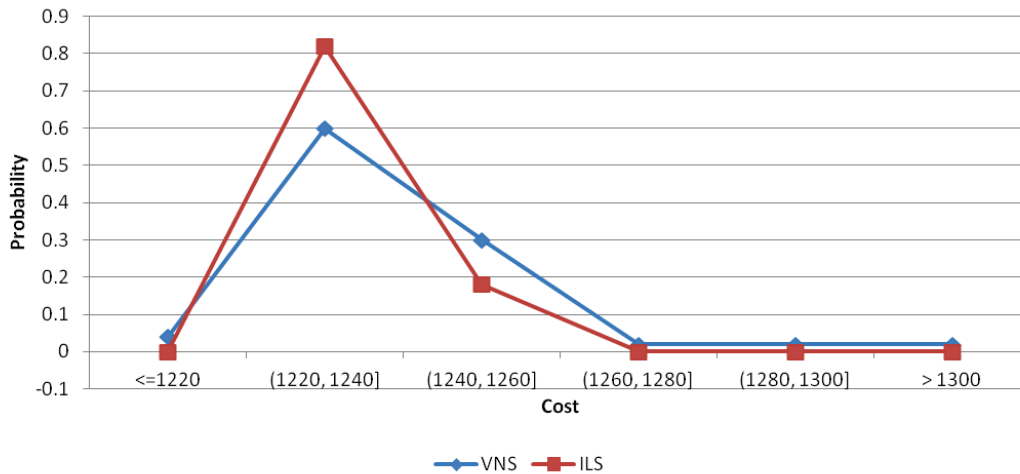


Figure 6.6: *Distribution of the output solution values for 50 simulation runs on instance C6 for a computational time limit of 300 seconds*

6.6-6.9. From the plots, the following observations can be made:

- In general, the mean of the output values for the VNS was found to be around the same or smaller than the ILS, implying that on an average, the VNS algorithm performs better than the ILS for most instances.
- There is a larger probability of finding a good solution using the VNS as compared to the ILS, as indicated by the frequency of the output solution values in the low cost range as shown in the figures.
- The VNS is however less stable than the ILS as evident from the concentration of the output solution values in a single output range for the ILS, as represented by the peak in the distribution curve for the ILS.

Thus for a given instance, the VNS algorithm is expected to perform better on an average with a higher probability of finding a good solution, but there is also a larger variance in the output solution values returned by the VNS algorithm.

## 6.6 Conclusions and Future Work

This study demonstrates the complexity in dealing with uncertainty in release times and processing times of jobs in a proactive manner for the most basic form of the machine scheduling problem. In our problem, the release times and

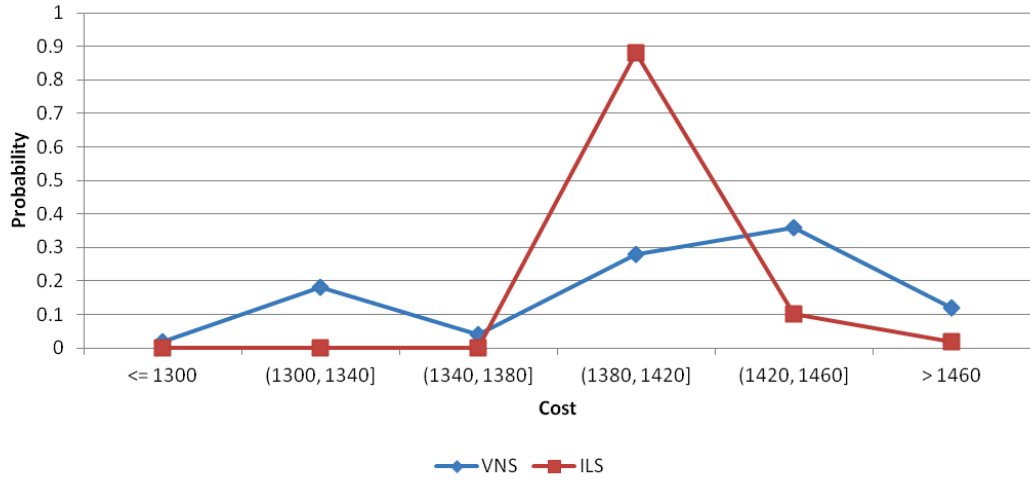


Figure 6.7: Distribution of the output solution values for 50 simulation runs on instance C12 for a computational time limit of 300 seconds

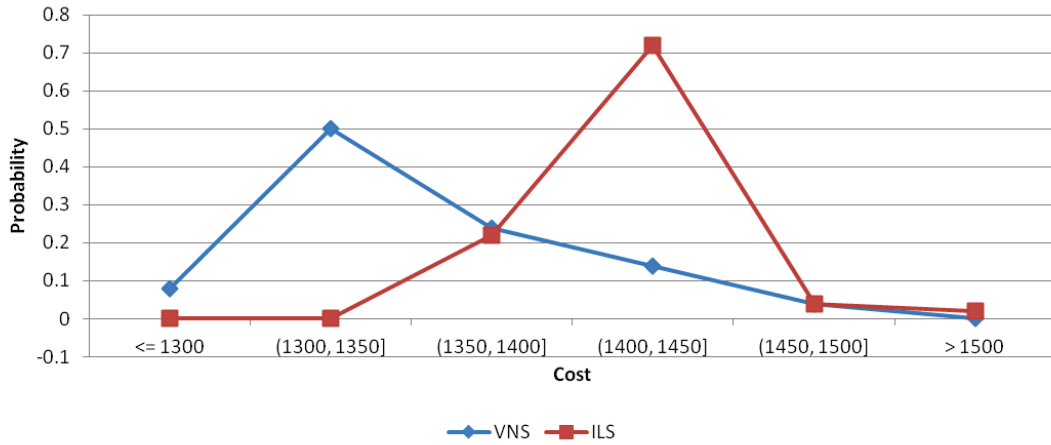


Figure 6.8: Distribution of the output solution values for 50 simulation runs on instance C16 for a computational time limit of 300 seconds

processing times of jobs are specified as independent ranges of values with unknown probability distributions. The performance criterion is the total flow time of all jobs and the robustness measure is the realized outcome for the worst-case contingency over the set of all possible scenarios. In previous research, the uncertainty in the release times of the jobs was largely ignored in the robust scheduling context. In this research, we illustrate the added complexity in considering release times, and show that in order to solve the absolute robust single machine scheduling problem for  $n$  jobs, we can restrict our attention to a subset of cardinality  $2^n$  of the extreme point scenarios from the set

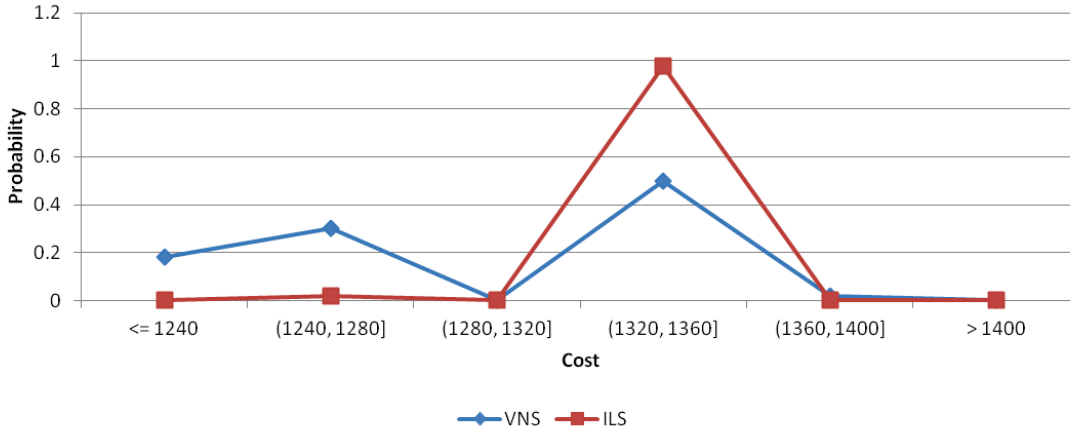


Figure 6.9: *Distribution of the output solution values for 50 simulation runs on instance C19 for a computational time limit of 300 seconds*

of infinite possible realizations of release times and processing times. We propose heuristic algorithms based on variable neighborhood search and iterated local search to generate schedules with the best performance in the worst case contingency. The variable neighborhood search (VNS) algorithm was able to solve all instances with  $|N|=7$  jobs to optimality. For larger problem size, on the average, the VNS was found to perform better than ILS with a larger associated probability of finding good solutions. However, the VNS was found to be less stable than the ILS as indicated by the variance in the output solution values.

In this chapter, we have investigated the problem of developing proactive strategies to handle uncertainty in the single machine scheduling problem. As part of future work, the proposed methodology for the single machine scheduling problem can be extended to more than one machine. There is further scope for research on developing robust schedules for the single machine scheduling problem with uncertainty in release times and processing times, with the robustness measure as the maximum regret with respect to the corresponding optimal solution over the set of all possible realizations. There can also be several other performance criteria such as the sum of completion times of all jobs or the total tardiness of all jobs beyond the specified due times for finishing.

## 6.6. Conclusions and Future Work

Table 6.2: Computational results for generated instances with  $|N|=7$

Instance	Lower Bound <sup>1</sup>	Optimal Solution <sup>2</sup>		VNS		ILS		% Gap <sup>3</sup>
		cost	time	cost	time <sup>4</sup>	cost	time <sup>5</sup>	
$\beta = 2$								
A1	188	212	4	212	5	212	2	11.32%
A2	189	204	4	204	7	217	100	7.35%
A3	192	200	4	200	13	207	100	4.00%
A4	168	180	5	180	7	192	100	6.67%
A5	189	198	4	198	4	203	100	4.55%
Mean								<b>6.78%</b>
$\beta = 3$								
A6	174	189	5	189	2	189	1	7.94%
A7	126	137	4	137	4	139	100	8.03%
A8	160	183	5	183	7	183	2	12.57%
A9	162	202	4	202	3	202	1	19.80%
A10	194	200	4	200	4	206	100	3.00%
Mean								<b>10.27%</b>
$\beta = 4$								
A11	135	173	5	173	7	175	100	21.97%
A12	146	154	4	154	10	154	76	5.19%
A13	120	133	4	133	5	133	4	9.77%
A14	180	209	4	209	83	213	100	13.88%
A15	121	151	5	151	13	151	8	19.87%
Mean								<b>14.14%</b>
$\beta = 6$								
A16	148	178	4	178	8	178	5	16.85%
A17	141	189	4	189	9	189	5	25.40%
A18	150	192	4	192	8	200	100	21.88%
A19	134	171	4	171	12	173	100	21.64%
A20	156	183	5	183	7	194	100	14.75%
Mean								<b>20.10%</b>

<sup>1</sup>The lower bound is the maximum optimal value over the set of all possible scenarios.

<sup>2</sup>The optimal solution is determined using the exhaustive search algorithm.

<sup>3</sup>The optimality gap of the lower bound with respect to the optimal solution.

<sup>4</sup>A computational time limit of 100 seconds was set for all instances with  $|N|=7$  jobs.

<sup>5</sup>A computational time limit of 100 seconds was set for all instances with  $|N|=7$  jobs.

Table 6.3: Computational results for generated instances with  $|N|=15$

Instance	Lower Bound	VNS		ILS	
		cost	time <sup>6</sup>	cost	time <sup>7</sup>
$\beta = 2$					
B1	726	761	300	761	300
B2	703	783	300	783	300
B3	748	783	300	783	300
B4	674	710	300	710	300
B5	713	750	300	809	300
$\beta = 3$					
B6	716	757	300	757	300
B7	654	690	300	747	300
B8	712	742	300	742	300
B9	630	657	300	665	300
B10	593	624	300	637	300
$\beta = 4$					
B11	599	701	600	723	600
B12	587	681	600	763	600
B13	601	698	600	713	600
B14	687	701	600	770	600
B15	620	676	600	778	600
$\beta = 6$					
B16	671	763	600	763	600
B17	663	723	600	723	600
B18	675	748	600	772	600
B19	689	771	600	841	600
B20	724	871	600	889	600

---

<sup>6</sup>The computational time limit determined from a trial based analysis.

<sup>7</sup>The computational time limit determined from a trial based analysis.

Table 6.4: Computational results for generated instances with  $|N|=20$ 

Instance	Lower Bound	VNS		ILS	
		cost	time <sup>8</sup>	cost	time <sup>9</sup>
$\beta = 2$					
C1	1254	1322	600	1322	600
C2	1305	1410	600	1410	600
C3	1289	1369	600	1434	600
C4	1259	1395	600	1399	600
C5	1259	1338	600	1338	600
$\beta = 3$					
C6	1117	1228	600	1228	600
C7	1226	1274	600	1364	600
C8	1237	1317	600	1365	600
C9	1206	1328	600	1328	600
C10	1144	1269	600	1356	600
$\beta = 4$					
C11	1131	1342	900	1329	900
C12	1094	1369	900	1390	900
C13	1208	1362	900	1363	900
C14	1084	1312	900	1312	900
C15	1130	1351	900	1426	900
$\beta = 6$					
C16	1063	1213	900	1225	900
C17	1038	1144	900	1259	900
C18	1085	1355	900	1363	900
C19	1067	1215	900	1337	900
C20	1116	1357	900	1357	900

---

<sup>8</sup>The computational time limit determined from a trial based analysis.

<sup>9</sup>The computational time limit determined from a trial based analysis.

Table 6.5: Computational results for generated instances with  $|N|=30$

Instance	Lower Bound	VNS		ILS	
		cost	time <sup>10</sup>	cost	time <sup>11</sup>
$\beta = 2$					
D1	2865	3252	600	3286	600
D2	2595	2985	600	3126	600
D3	2768	3063	600	3120	600
D4	2647	3095	600	2982	600
D5	2799	3328	600	3335	600
$\beta = 3$					
D6	2526	2871	600	2786	600
D7	2452	3170	600	3183	600
D8	2525	2946	600	3160	600
D9	2673	3147	600	3154	600
D10	2809	3263	600	3326	600
$\beta = 4$					
D11	2378	2870	900	3085	900
D12	2543	3224	900	3320	900
D13	2294	2612	900	2593	900
D14	2551	3135	900	3193	900
D15	2362	2826	900	2835	900
$\beta = 6$					
D16	2296	3004	900	3092	900
D17	2224	2836	900	2748	900
D18	2205	3262	900	3320	900
D19	2190	3082	900	3256	900
D20	2153	2892	900	2914	900

---

<sup>10</sup>The computational time limit determined from a trial based analysis.

<sup>11</sup>The computational time limit determined from a trial based analysis.



Table 6.6: Computational results for generated instances with  $|N|=50$ 

Instance	Lower Bound	VNS		ILS	
		cost	time <sup>12</sup>	cost	time <sup>13</sup>
$\beta = 2$					
E1	7761	9326	900	9363	900
E2	7608	9060	900	9245	900
E3	7650	9001	900	8989	900
E4	7655	8670	900	8910	900
E5	7650	9568	900	9574	900
$\beta = 3$					
E6	7423	8718	900	9127	900
E7	7057	8129	900	8675	900
E8	7337	8616	900	8642	900
E9	7846	9111	900	9185	900
E10	7135	8710	900	8722	900
$\beta = 4$					
E11	7647	9500	1200	9767	1200
E12	7672	9642	1200	9655	1200
E13	7099	8836	1200	8848	1200
E14	7478	8972	1200	9207	1200
E15	7354	9126	1200	9321	1200
$\beta = 6$					
E16	6861	8598	1200	8596	1200
E17	7062	9105	1200	9116	1200
E18	7309	8662	1200	8671	1200
E19	6984	8677	1200	8671	1200
E20	7076	8630	1200	8957	1200

---

<sup>12</sup>The computational time limit determined from a trial based analysis.

<sup>13</sup>The computational time limit determined from a trial based analysis.



## 7 Conclusions and Future Work

In this chapter, we review the main results of this thesis, and discuss perspectives for future research.

### 7.1 Conclusions

The existing operations research (OR) literature on port operations planning almost entirely focuses on container terminal management. In *Part I* of the thesis, we develop innovative models and algorithms for bulk terminal management, emphasizing on the similarities and differences in applications and methodologies across the domains of container terminals and bulk ports.

In Chapter 2 of the thesis, we do a comparative analysis of the decision problems arising in bulk ports and container terminals from an OR perspective. We provide a brief survey of the existing literature on container terminals with a two-fold objective. On the one hand we study to what extent the existing models and algorithms for container terminals can be extended to bulk ports. On the other hand we identify specific bulk port features and discuss open research problems in the bulk context that are promising for future research.

In Chapter 3, we study the deterministic berth allocation problem (BAP) in bulk ports, explicitly considering the cargo type on the vessel and the locations of fixed specialized facilities such as conveyors and pipelines that are installed at certain sections along the quay. We present two alternative exact solution methods based on mixed integer programming and generalized set partitioning, and a meta-heuristic based on the principle of squeaky wheel optimization (SWO) to solve the problem. The results look promising and indicate that the proposed

set partitioning algorithm and the SWO meta-heuristic can be successfully used to solve realistic sized instances containing up to 40 vessels. The research presented in the chapter makes a significant contribution to the existing literature, as this is the first study that solves the BAP in the context of bulk ports.

Chapter 4 of this thesis proposes an exact solution algorithm based on the branch-and-price framework to solve the integrated problem of berth allocation and yard assignment in bulk ports. In the proposed model, the master problem is formulated as a set-partitioning problem, and in each iteration of the column generation process, a single sub-problem for each berthing vessel is solved separately using mixed integer programming. Results based on instances inspired from real bulk port data suggest that the proposed algorithm can be used to solve realistic sized instances within a reasonable computational time. To the best of our knowledge, this is the first study that proposes an exact method to solve the combined large scale problem of berth allocation and yard assignment in the field of port operations planning, since all the previous studies in the context of container terminals use meta-heuristic approaches to solve the problem.

*Part II* of the thesis is devoted to the study and development of innovative solution methods and techniques to handle uncertainty in scheduling problems.

In Chapter 5, we address the real-world problem of reacting to disruptions in real-time in the berth allocation of vessels at a port. We solve the BAP on a rolling planning horizon with the objective to minimize the total realized costs of berthing the vessels as the actual arrival and handling time data is revealed in real time. The uncertainty in the available information is modeled using probability distributions derived from past data. We propose a recovery method based on the re-optimization of the berthing assignment of the unassigned vessels in the events of disruption, and an alternative smart greedy approach to re-assign vessels. The proposed methodology to model uncertainty and recovery algorithms are tested and validated on instances inspired from real port data. The results indicate that the total realized costs of berthing vessels can be significantly reduced as compared to the ongoing practice of reassigning vessels at the port. In previous research in port operations planning, reactive strategies to handle uncertainty are based on either simple rules of thumb or local heuristics. Thus the research presented in this chapter makes an important contribution in the context of developing sophisticated reaction-based strategies to handle uncertainty in port operations.

Chapter 6 of the thesis focuses on handling uncertainty in a proactive manner for the general scheduling problem. To illustrate the complexity in using a proactive approach and permit elegant and useful analysis, we consider the simplest scheduling problem - the single machine scheduling problem, in which the release times and the processing times of the jobs are specified as independent ranges of values with unknown probability distributions. The performance criterion is the total flow time of all jobs, and the measure of robustness is the worst-case performance from the set of all potential scenarios of release times and processing times. We demonstrate the added complexity of considering non-zero release times that was largely ignored in the past literature on robust scheduling, and provide important insights in developing robust schedules for the single machine scheduling problem. We further propose heuristic algorithms based on variable neighborhood search and iterated local search to generate robust schedules.

## 7.2 Future Research Perspectives

The research presented in this thesis attempts to answer several open research questions in the field of port operations planning and scheduling, and points out many more problems worthy of investigation in future research. In Chapter 2 of the thesis, we have demonstrated that there is tremendous scope for future research in bulk terminal management. We discuss the combined stowage planning and vessel routing problem with the objective to minimize the total travel time or the total distance to serve the demand at a given set of ports by a fleet of vessels. The routes should be designed such that the vessel stability requirements and the draft restrictions are met at each port of rotation. As discussed, the problem is particularly challenging in the bulk context where it is usually not possible to store multiple brands of cargo in the same hold of a given vessel.

Another challenging problem is to extend the berth allocation model proposed by Umang et al. (2013) and account for the integrated planning of berth allocation and crane scheduling in the bulk context. As discussed in Chapter 2, the problem differs from that in container terminals, as it is necessary to account for the cargo type on the vessel, the locations of the specialized equipment facilities such as conveyors and pipelines and the redundancy of the non-interference constraints for the mobile harbor cranes used in bulk ports.

There is currently no existing literature on the deployment, routing and scheduling of equipment for transfer operations and yard management in bulk ports,

and thus there are several open research problems in this area. In particular, improvements in the design, operations and deployment of conveyors and pipelines represents a promising direction for future research. Some of these decision problems can also be integrated and solved as large scale optimization problems, such as the combined problem of berth allocation and deployment of transfer equipment with or without integration with yard assignment.

In Chapter 3, the proposed integrated model for berth allocation and yard assignment in bulk ports assumes that each berthing vessel loads or discharges a single cargo type. The model can be extended to vessels carrying multiple cargo types, including vessels that load and discharge cargo at the same port, by adding extra constraints and decision variables. On the methodological front, a more in-depth understanding of sophisticated techniques such as dual stabilization and dynamic constraint aggregation is needed to improve the efficiency of the proposed exact method based on the branch-and-price framework. The overall efficiency of the proposed exact model can be further improved by reducing the computational time for solving the sub-problems by using meta-heuristics or dynamic programming instead of solving them to optimality using optimization solvers.

Another open research problem in the field of port operations planning is to devise pricing strategies that can enable the port to earn revenue from the late-arriving vessels and other port entities including trucking companies and cargo agents to partially or wholly recover the cost of modifying their planned schedules in response to disruptions. This is a complex issue because typically in a port system, there are several stakeholders including the port authorities, shipping companies, trucking companies, cargo agents and others, who may often have conflicting interests and objectives. Thus for any revenue maximizing policy to be put into practice and successfully implemented, it should be fair and beneficial to all the stakeholders. We believe that a game-theoretical approach in this context would be an interesting and challenging direction for future research.

The existing literature on handling uncertainty in scheduling problems, particularly in the context of port operations planning is very scarce. In this thesis, we have touched upon the problem of developing proactive strategies to handle uncertainty in scheduling problems. The proposed methodology for the single machine scheduling problem can be extended to more than one machine. There can be also other robustness criteria such as the minimization of maximum regret over the set of all possible realizations, and alternative per-

formance measures such as the sum of the completion times or tardiness of all jobs. A key methodological challenge is in developing robust planning algorithms that ensure a minimum desired level of service for a certain anticipated level of uncertainty in the available information. We believe that more sophisticated techniques such as the robust optimization approach of Bertsimas and Sim (2004) can be used to overcome this challenge.





## Bibliography

- Albers, S. (2003). Online algorithms: A survey, *Mathematical Programming* **97**: 3–26. Invited paper at ISMP 2003.
- Alvelos, F. (2005). *Branch-and-Price and Multicommodity Flows*, PhD thesis, Universidade do Minho.
- Ambrosino, D., Sciomachen, A. and Tanfani, E. (2006). A decomposition heuristics for the container ship stowage problem, *Journal of Heuristics* **12**: 211–233.
- Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P. and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs, *Operations Research* **46**(3): 316–329.
- Bean, J., Birge, J., Mittenthal, J. and Noon, C. (1987). Matching-up scheduling with multiple resources, release dates and disruptions, *Operations Research* **39** (3): 470–483.
- Ben-Tal, A. and Nemirovski, A. (1998). Robust convex optimization, *Mathematics of Operations Research* **23**(4).
- Ben-Tal, A. and Nemirovski, A. (1999). Robust solutions of uncertain linear programs, *OR Letters* **25**: 1–13.
- Ben-Tal, A. and Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data, *Mathematical Programming* **88**: 414–424.
- Bertsimas, D. and Sim, M. (2003). Robust discrete optimization, *Mathematical Programming* **98**: 49–71.
- Bertsimas, D. and Sim, M. (2004). The price of robustness, *Operations Research* **52**: 35–53.

## Bibliography

---

- Bierwirth, C. and Meisel, F. (2009). A fast heuristic for quay crane scheduling with interference constraints, *Journal of Scheduling* **12**(4): 345–360.
- Bierwirth, C. and Meisel, F. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals, *European Journal of Operational Research* **202**(3): 615–627.
- Birge, J. R. and Louveaux, F. (eds) (1997). *Introduction to Stochastic Programming*, Springer.
- Bish, E. K., Leong, T. Y., Li, C. L., Ng, J. W. C. and Simchi-Levi, D. (2001). Analysis of a new vehicle scheduling and location problem, *Naval Research Logistics* **48**: 363–385.
- Blazewicz, J. (1987). Selected topics in scheduling theory, *Annals of Discrete Mathematics* **31**: 1–60.
- Boland, N., Gulczynski, D. and Savelsbergh, M. W. P. (2012). A stockyard planning problem, *Euro Journal on Transportation and Logistics* **1**: 197–236.
- Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J. and Lusby, R. (2011). Models for the discrete berth allocation problem: A computational comparison, *Transportation Research Part E* **47**(4): 461 – 473.
- Cai, X. and Zhou, X. (2005). Single-machine scheduling with exponential processing times and general stochastic cost functions, *Journal of Global Optimization* **31**: 317–332.
- Chang, D., Yan, W., Chen, C.-H. and Jiang, Z. (2008). A berth allocation strategy using heuristics algorithm and simulation optimisation, *Int. J. Comput. Appl. Technol.* **32**: 272–281.
- Chen, J. H., Lee, D. H. and Mark, G. (2014). An effective mathematical formulation for the unidirectional cluster-based quay crane scheduling problem, *European Journal of Operational Research* **232**: 198–208.
- Chen, Z. L., Lei, L. and Zhong, H. (2007). Container vessel scheduling with bi-directional flows, *Operations Research Letters* **35**: 186–194.
- Cheng, Y. L., Sen, H. C., Natarajan, K., Teo, C. P. and Tan, K. C. (2005). Dispatching automated guided vehicles in a container terminal, in J. Geunes and P. M. Pardalos (eds), *Supply Chain Optimization, Applied Optimization*, Springer.

- Cheong, C., Tan, K., Liu, D. and Lin, C. (2010). Multi-objective and prioritized berth allocation in container ports, *Annals of Operations Research* **180**: 63–103.
- Cheung, R. K., Li, C. L. and Lin, W. (2002). Interblock crane deployment in container terminals, *Transportation Science* **36**: 79–93.
- Christensen, C. G. and Holst, C. T. (2008). *Berth allocation in container terminals*, Master's thesis, Technical University of Denmark.
- Clements, D., Crawford, J., Joslin, D., Nemhauser, G. L., Puttlitz, M. and Savelsbergh, M. W. P. (1997). Heuristic optimization: A hybrid ai/or approach, *Proceedings of the Workshop on Industrial Constraint-Directed Scheduling*.
- Cordeau, J. F., Gaudioso, M., Laporte, G. and Moccia, L. (2006). A memetic heuristic for the generalized quadratic assignment problem, *INFORMS Journal on Computing* **18**(4): 433–443.
- Cordeau, J. F., Gaudioso, M., Laporte, G. and Moccia, L. (2007). The service allocation problem at the Gioia Tauro maritime terminal, *European Journal of Operational Research* **176**(2): 1167–1184.
- Cordeau, J. F., Laporte, G., Legato, P. and Moccia, L. (2005). Models and tabu search heuristics for the berth-allocation problem, *Transportation Science* **39**(4): 526–538.
- Crainic, T. G., Gendreau, M. and Dejax, P. (1993). Dynamic and stochastic models for the allocation of empty containers, *Operations Research* **41**: 102–126.
- Daganzo, C. F. (1989). The crane scheduling problem, *Transportation Research Part B* **23**: 159–175.
- Dai, J., Lin, W., Moorthy, R. and Teo, C. P. (2008). Berth allocation planning optimization in container terminals, in C. S. Tang, C. P. Teo and K. K. Wei (eds), *Supply Chain Analysis*, Vol. 119 of *International Series in Operations Research & Management Science*, Springer, pp. 69–104.
- Daniels, R. and Kouvelis, P. (1995). Robust scheduling to hedge against processing time uncertainty in single-stage production, *Management Science* **41** (2): 363–376.
- Desaulniers, G., Desrosiers, J. and Solomon, M. (eds) (2005). *Column Generation*, GERAD 25th Anniversary Series, Springer.

## Bibliography

---

- Du, Y., Xu, Y. and Chen, P. (2010). A feedback procedure for robust berth allocation with stochastic vessel delays, *Proceedings of the 8th World Congress on Intelligent Control and Automation*, Jinan, China.
- Elhallaoui, I., Desaulniers, G., Metrane, A. and Soumis, F. (2008). Bi-dynamic constraint aggregation and subproblem reduction, *Comput. Oper. Res.* **35**(5): 1713–1724.  
URL:<http://dx.doi.org/10.1016/j.cor.2006.10.007>
- Elhallaoui, I., Villeneuve, D., Soumis, F. and Desaulniers, G. (2005). Dynamic aggregation of set-partitioning constraints in column generation, *Oper. Res.* **53**(4): 632–645.  
URL:<http://dx.doi.org/10.1287/opre.1050.0222>
- Emmons, H. and Pinedo, M. (1990). Scheduling stochastic jobs with due dates on parallel machines, *European Journal of Operational Research* **47**: 49–55.
- Erera, A. L., Morales, J. C. and Savelsbergh, M. W. P. (2005). Global intermodal tank container management for the chemical industry, *Transportation Research Part E* **41**: 551–566.
- Erera, A. L., Morales, J. C. and Savelsbergh, M. W. P. (2009). Robust optimization for empty repositioning problems, *Operations Research* **57**: 468–483.
- Feillet, D. (2010). A tutorial on column generation and branch-and-price for vehicle routing problems, *4or* **8**(4): 407–424.  
URL:<http://www.springerlink.com/index/10.1007/s10288-010-0130-z>
- Fu, Z., Li, C. L., Lim, A. and Rodrigues, B. (2007). Port space allocation with a time dimension, *JORS* **58**: 797–807.
- Gao, C., Zhang, R., Du, Y. and Chen, Q. (2010). A proactive and reactive framework for berth allocation with uncertainties, *Advanced Management Science (ICAMS), 2010 IEEE International Conference on*, Vol. 3, pp. 144–149.
- Garey, M. R. and Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*, Freeman & co., New York.
- Giallombardo, G., Moccia, L., Salani, M. and Vacca, I. (2010). Modeling and solving the tactical berth allocation problem, *Transportation Research Part B* **44**(2): 232–245.
- Gittins, J. and Glazebrook, K. (1977). On bayesian models in stochastic scheduling, *Journal of Applied Probabilities* **14**: 556–565.

- Glazebrook, K. (1979). Scheduling tasks with exponential service times on parallel processors, *Journal of Applied Probability* **16**: 685–689.
- Glazebrook, K. (1981). On non-preemptive strategies in stochastic scheduling, *Naval Research Logistics Quarterly* **28**: 289–300.
- Glazebrook, K. (1985). Semi-markov models for single machine stochastic scheduling problems, *International Journal of Systems Science* **16 (5)**: 573–587.
- Graham, R., Lawler, E., Lenstra, J. and Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey, *Annals of Discrete Mathematics* **5**: 287–326.
- Guan, Y. and Cheung, R. K. (2004). The berth allocation problem: models and solution methods, *OR Spectrum* **26**: 75–92.
- Guan, Y., Xiao, W.-Q., Cheung, R. K. and Li, C.-L. (2002). A multiprocessor task scheduling model for berth allocation: heuristic and worst-case analysis, *Operations Research Letters* **30(5)**: 343 – 350.
- Hahn, P. M., Kim, B.-J., Guignard, M., Smith, J. M. and Zhu, Y.-R. (2008). An algorithm for the generalized quadratic assignment problem, *Comput. Optim. Appl.* **40(3)**: 351–372.  
URL:<http://dx.doi.org/10.1007/s10589-007-9093-1>
- Han, X.-l., Lu, Z.-q. and Xi, L.-f. (2010). A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time, *European Journal of Operational Research* **207(3)**: 1327 – 1340.
- Hansen, P. and Mladenovic, N. (1997). Variable neighborhood search, *Computers and Operations Research* **24**: 1097–1100.
- Herroelen, W. and Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials, *European Journal of Operational Research* **165 (2)**: 289–306.
- Hillier, F. S. and Lieberman, G. J. (eds) (2001). *Introduction to Operations Research*, McGraw-Hill Science/Engineering/Math.
- Imai, A., Chen, H. C., Nishimura, E. and Papadimitriou, S. (2008). The simultaneous berth and quay crane allocation problem, *Transportation Research Part E* **44(5)**: 900–920.

## Bibliography

---

- Imai, A., Nagaiwa, K. and Chan, W. T. (1997). Efficient planning of berth allocation for container terminals in Asia, *Journal of Advanced Transportation* **31**(1): 75–94.
- Imai, A., Nishimura, E., Hattori, M. and Papadimitriou, S. (2007). Berth allocation at indented berths for mega-containerships, *European Journal of Operational Research* **179**(2): 579–593.
- Imai, A., Nishimura, E. and Papadimitriou, S. (2001). The dynamic berth allocation problem for a container port, *Transportation Research Part B* **35**(4): 401–417.
- Imai, A., Nishimura, E. and Papadimitriou, S. (2003). Berth allocation with service priority, *Transportation Research Part B* **37**(5): 437–457.
- Imai, A., Nishimura, E. and Papadimitriou, S. (2008). Berthing ships at a multi-user container terminal with a limited quay capacity, *Transportation Research Part E* **44**: 136–151.
- Imai, A., Sasaki, K., Nishimura, E. and Papadimitriou, S. (2006). Multi-objective simultaneous stowage and load planning for a container ship with container rehandle in yard stacks, *European Journal of Operational Research* **171**: 373–389.
- Imai, A., Sun, X., Nishimura, E. and Papadimitriou, S. (2005). Berth allocation in a container port: using a continuous location space approach, *Transportation Research Part B* **39**(3): 199–221.
- Jenkins, C. (2009). *A ship routing problem with hold and draft constraints*, Master's thesis, The University of Melbourne.
- Joslin, D. and Clements, D. (1998). "squeaky wheel" optimization, *AAAI'98 Proceedings of the national conference on Artificial intelligence*.
- Jung, S. H. and Kim, K. H. (2006). Load scheduling for multiple quay cranes in port container terminals, *Journal of Intelligent Manufacturing* **17**: 479–492.
- Kall, P. and Mayer, J. (eds) (2005). *Stochastic Linear Programming, Models, Theory and Computation*, Springer.
- Kasperski, A. (2005). Minimizing maximal regret in the single machine sequencing problem with maximum lateness criterion, *Operations Research Letters* **33**: 431–436.



- Kellerer, H., Tautenhahn, T. and Woeginger, G. (1999). Approximability and nonapproximability results for minimizing total flow time on a single machine, *SIAM Journal of Computing* **28**(4): 1155–1166.
- Kim, K. H. and Kim, K. Y. (1999). Routing straddle carriers for the loading operation of containers using a beam search algorithm, *Computers and Industrial Engineering* **36**(1): 109–136.
- Kim, K. H., Lee, K. M. and Hwang, H. (2003). Sequencing delivery and receiving operations for yard cranes in port container terminals, *International Journal of Production Economics* **84**: 283–292.
- Kim, K. H. and Moon, K. C. (2003). Berth scheduling by simulated annealing, *Transportation Research Part B* **37**(6): 541–560.
- Kim, K. H. and Park, Y. M. (2004). A crane scheduling method for port container terminals, *European Journal of Operational Research* **156**(3): 752–768.
- Kouvelis, P. and Yu, G. (1997). *Robust Discrete Optimization and Its Applications*, Kluwer Academic Publishers, Dordrecht, Germany.
- Kozan, E. and Preston, P. (2006). Mathematical modelling of container transfers and storage locations at seaport terminals, *OR Spectrum* **28**: 519–537.
- Lau, H. and Zhao, Y. (2007). Integrated scheduling of handling equipment at automated container terminals, *International Journal of Production Economics* p. doi:10.1016/j.ijpe.2007.05.015.
- Lawler, E. (1976). *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York.
- Lee, D. H. and Jin, J. (2013). Feeder vessel management at container transshipment terminals, *Transportation Research Part E* **49**: 201–216.
- Lee, L. H., Chew, E. P., Tan, K. C. and Han, Y. (2006). An optimization model for storage yard management in transshipment hubs, *OR Spectrum* **28**: 539–561.
- Lenstra, J., Kan, A. R. and Brucker, P. (1977). Complexity of machine scheduling problems, *Annals of Discrete Mathematics* pp. 343–362.
- Li, C. L., Cai, X. and Lee, C. Y. (1998). Scheduling with multiple-job-on-one-processor pattern, *IIE Transactions* **30**: 433–445.

## Bibliography

---

- Lim, A. (1998). The berth planning problem, *Operations Research Letters* **22**(2-3): 105–110.
- Lim, A., Rodrigues, B., Xiao, F. and Zhu, Y. (2004). Crane scheduling with spatial constraints, *Naval Research Logistics* **51**: 386–406.
- Lim, A. and Xu, Z. (2006). A critical-shaking neighborhood search for the yard allocation problem, *European Journal of Operational Research* **174**(2): 1247–1259.
- Liu, C. I., Jula, H., Vukadinovic, K. and Ioannou, P. (2004). Automated guided vehicle system for two container yard layouts, *Transportation Research Part C* **12**: 349–368.
- Lu, C., Lin, S. and Ying, K. (2012). Robust scheduling on a single machine to minimize total flow time, *Computers & Operations Research* **39**: 1682–1691.
- Mauri, G., Oliveira, A. and Lorena, L. (2008). A hybrid column generation approach for the berth allocation problem, in J. van Hemert and C. Cotta (eds), *Evolutionary Computation in Combinatorial Optimization*, Vol. 4972 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 110–122.
- Meisel, F. and Bierwirth, C. (2006). Integration of berth allocation and crane assignment to improve the resource utilization at a seaport container terminal, *Operations Research Proceedings 2005*, Springer, pp. 105–110.
- Meisel, F. and Bierwirth, C. (2009). Heuristics for the integration of crane productivity in the berth allocation problem, *Transportation Research Part E* **45**(1): 196–209.
- Moccia, L., Cordeau, J. F., Gaudioso, M. and Laporte, G. (2006). A branch-and-cut algorithm for the quay crane scheduling problem in a container terminal, *Naval Research Logistics* **53**(1): 45–59.
- Mohring, R., Radermacher, F. and Weiss, G. (1984). Stochastic scheduling problems i: General strategies, *Mathematical Methods of Operations Research (ZOR)* **28**: 193–260.
- Mohring, R., Radermacher, F. and Weiss, G. (1985). Stochastic scheduling problems ii: Set strategies, *Mathematical Methods of Operations Research (ZOR)* **29**: 65–104.



- Monaco, M. F. and Sammarra, M. (2007). The berth allocation problem: a strong formulation solved by a lagrangean approach, *Transportation Science* **41**(2): 265–280.
- Moorthy, R. and Teo, C. P. (2006). Berth management in container terminal: the template design problem, *OR Spectrum* **28**(4): 495–518.
- Ng, J. W. C. (2005). Crane scheduling in container yards with inter-crane interference, *European Journal of Operational Research* **164**: 64–78.
- Ng, J. W. C. and Mak, K. L. (2005). Yard crane scheduling in port container terminals, *Applied Mathematical Modelling* **29**: 263–276.
- Nishimura, E., Imai, A. and Papadimitriou, S. (2001). Berth allocation planning in the public berth system by genetic algorithms, *European Journal of Operational Research* **131**(2): 282 – 292.
- Nishimura, E., Imai, A. and Papadimitriou, S. (2009). Container storage and transshipment marine terminals, *European Journal of Operational Research* **45**: 771–786.
- Park, K. T. and Kim, K. H. (2002). Berth scheduling for container terminals by using a sub-gradient optimization technique, *Journal of the Operational Research Society* **53**: 1054–1062.
- Park, Y. M. and Kim, K. H. (2003). A scheduling method for berth and quay cranes, *OR Spectrum* **25**(1): 1–23.
- Phillips, C., Stein, C. and Wein, J. (1998). Minimizing average completion time in the presence of release dates, *Mathematical Programming* **82**: 199–223.
- Pigatti, A., Poggi and Uchoa, E. (2005). Stabilized Branch-and-cut-and-price for the Generalized Assignment Problem, *GRACO'05*, Vol. 19, pp. 389–395.
- Pinedo, M. (1983). Stochastic scheduling with release dates and due dates, *Operations Research* **31**: 559–572.
- Pinedo, M. (2002). *Scheduling: Theory, Algorithms, and Systems*, Prentice Hall, Englewood Cliffs.
- Pinedo, M. and Schrage, L. (1982). Stochastic shop scheduling: A survey, In M.A.H. Dempster, J.K. Lenstra, A.H.G. Rinnooy Kan (Eds.), *Deterministic and Stochastic Scheduling*, Rediel, Dordrecht **84**: 181–196.

## Bibliography

---

- Robenek, T., Umang, N., Bierlaire, M. and Ropke, S. (2014). A branch-and-price algorithm to solve the integrated berth allocation and yard assignment problem in bulk ports, *European Journal of Operational Research* **235**(2): 399–411.
- Roundy, R., Herer, Y. and Tayur, S. (1989). Price-directed scheduling of production operations in real time, *Proc. 15th Conference of the NSF Production Research and Technology Program*.
- Sammarra, M., Cordeau, J. F., Laporte, G. and Monaco, M. F. (2007). A tabu search heuristic for the quay crane scheduling problem, *Journal of Scheduling* **10**: 327–336.
- Sciomachen, A. and Tanfani, E. (2007). A 3d-bpp approach for optimising stowage plans and terminal productivity, *European Journal of Operational Research* **183**: 1433–1446.
- Shintani, K., Imai, A., Nishimura, E. and Papadimitriou, S. (2007). The container shipping network design problem with empty container repositioning, *Transportation Research Part E* **43**: 39–59.
- Singh, G., Sier, D., Ernst, A. T., Gavrilouk, O., Oyston, R., Giles, T. and Welgama, P. (2012). A mixed integer programming model for long term capacity expansion planning: A case study from the hunter valley coal chain, *European Journal of Operational Research* **220**: 210–224.
- Skutella, M. and Uetz, M. (2005). Stochastic machine scheduling with precedence constraints, *SIAM Journal on Computing* **34**: 788–802.
- Smith, T. B. and Pyle, J. M. (2004). An effective algorithm for project scheduling with arbitrary temporal constraints, *AAAI'04 Proceedings of the 19th national conference on Artificial intelligence*, pp. 544–549.
- Smith, W. (1956). Various optimizers for single-stage production, *Naval Research Logistics* **3**: 59–66.
- Soroush, H. and Fredendall, L. (1994). The stochastic single machine scheduling problem with earliness and tardiness costs, *European Journal of Operational Research* **77**: 287–302.
- Soyster, A. (1973). Convex programming with set-inclusive constraints and application to inexact linear programming, *Operations Research* **21**(5): 1154–1157.

- Stahlbock, R. and Voss, S. (2008). Operations research at container terminals: a literature update, *OR Spectrum* **30**(1): 1–52.
- Steenken, D., Voss, S. and Stahlbock, R. (2004). Container terminal operation and operations research - a classification and literature review, *OR Spectrum* **26**(1): 3–49.
- Tong, C., Lau, H. and Lim, A. (1999). Ant colony optimization for the ship berthing problem, in P. Thiagarajan and R. Yap (eds), *ASIAN '99 Proceedings of the 5th Asian Computing Science Conference on Advances in Computer Science*, Vol. 1742 of *Lecture Notes in Computer Science*, Springer, pp. 789–789.
- Umang, N., Bierlaire, M. and Vacca, I. (2013). Exact and heuristic methods to solve the berth allocation problem in bulk ports, *Transportation Research Part E* **54**: 14–31.
- UNCTAD (2012). *Review of Maritime Transport*, United Nations Conference on Trade and Development, <http://www.unctad.org>.
- Vacca, I. (2011). *Container terminal management: integrated models and large-scale optimization algorithms*, PhD thesis, Ecole Polytechnique Fédérale de Lausanne.
- Vis, I. F. A., de Koster, R. and Savelsbergh, M. W. P. (2005). Minimum vehicle fleet size under time-window constraints at a container terminal, *Transportation Science* **39**: 249–260.
- Wallace, S. and Ziemba, W. (1997). *Applications of stochastic programming, (Mps-Siam Series on Optimization)*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
- Watters, L. (1967). Reduction of integer polynomial programming problems to zero-one linear programming problems, *Operations Research* **15**: 1171–1174.
- Weiss, G. and Pinedo, M. (1980). Scheduling tasks with exponential service times on nonidentical parallel processors to minimize various cost functions, *Journal of Applied Probability* **17**: 187–202.
- Wu, X. and Zhou, X. (2008). Stochastic scheduling to minimize expected maximum lateness, *European Journal of Operational Research* **190**: 103–115.
- Xu, Y., Chen, Q. and Quan, X. (2012). Robust berth scheduling with uncertain vessel delay and handling time, *Annals of Operations Research* **192**: 123–140.

## Bibliography

---

- Yang, J. and Yu, G. (2002). On the robust single machine scheduling problem, *Journal of Combinatorial Optimization* **6**: 17–33.
- Zeng, Q., Yang, Z. and Hu, X. (2012). Disruption recovery model for berth and quay crane scheduling in container terminals, *Engineering Optimization* **43**: 967–983.
- Zhang, C., Wan, Y. W., Liu, J. and Linn, R. (2002). Dynamic crane deployment in container storage yards, *Transportation Research Part B* **36**: 537–555.
- Zhen, L. and Chang, D. F. (2012). A bi-objective model for robust berth allocation scheduling, *Computers and Industrial Engineering* **63**: 262–273.
- Zhen, L., Chew, E. P. and Lee, L. H. (2011). An integrated model for berth template and yard template planning in transshipment hubs, *Transportation Science* **45**: 483–504.
- Zhen, L., Lee, L. H. and Chew, E. P. (2011). A decision model for berth allocation under uncertainty, *European Journal of Operational Research* **212**: 54–68.
- Zhou, P.-f. and Kang, H.-g. (2008). Study on berth and quay-crane allocation under stochastic environments in container terminal, *Systems Engineering - Theory & Practice* **28**(1): 161 – 169.

# Nitish Umang

E-Mail: [nitish.umang@epfl.ch](mailto:nitish.umang@epfl.ch), [nitishumang86@gmail.com](mailto:nitishumang86@gmail.com)

Address : Chemin de la Prairie 62, Apartment 130, 2<sup>nd</sup> floor, 1007 Lausanne, Switzerland

Contact Number: +41 76 7874718, +41 21 6939327

## EDUCATION

---

### École Polytechnique Fédérale de Lausanne, Switzerland

Ph.D. in Operations Research

April 2014

- Dissertation: From container terminals to bulk ports: models and algorithms for integrated planning and robust scheduling
- Advisor: Prof. Michel Bierlaire
- Thesis Committee: Michel Bierlaire (EPFL), Alan Erera (Georgia Tech), Jean-Francois Cordeau (HEC Montréal), Daniel Kuhn (EPFL)

### Massachusetts Institute of Technology, Cambridge, MA, USA

Master of Science in Transportation

June 2010

Advisor: Prof. Amedeo Odoni

### Indian Institute of Technology Delhi, New Delhi, India

Bachelor of Technology in Civil Engineering with Minor in Computational Mechanics

May 2008

## RESEARCH INTERESTS

---

Mathematical modelling, large-scale optimization, meta-heuristics, robust scheduling and real-time re-planning for large-scale transportation and logistics systems

## WORK EXPERIENCE

---

### Georgia Institute of Technology, Atlanta, GA, USA

January – July 2013

Visiting Researcher, School of Industrial and Systems Engineering

Worked with Prof. Alan Erera on robust scheduling problems

### Swiss Federal Institute of Technology, Lausanne, Switzerland

Oct 2010 – March 2011

Research Intern, Transport and Mobility Laboratory

Worked with Prof. Michel Bierlaire and Dr. Ilaria Vacca on bulk port operations planning

## SELECTED PUBLICATIONS

---

- **Umang, N.**, Bierlaire, M., and Vacca, I. (2013). Exact and heuristic methods to solve the berth allocation problem in bulk ports. *Transportation Research Part E: Logistics and Transportation Review*, 54: 14–31.
- Robenek, T., **Umang, N.**, Bierlaire, M. And Ropke, S. (2014). A branch-and-price algorithm to solve the integrated berth allocation and yard assignment problem in bulk ports. *European Journal of Operational Research*, 235(2): 399–411.
- **Umang, N.**, Bierlaire, M. and Erera, A.L. (2013). Real-time management of the berth allocation problem with stochastic arrival and handling times. Technical report, TRANSP-OR, École polytechnique fédérale de Lausanne.
- **Umang, N.**, Erera, A.L. and Bierlaire, M. (2013). The robust single machine scheduling problem with uncertain release and processing times. Technical report, TRANSP-OR, École polytechnique fédérale de Lausanne.

# Nitish Umang

E-Mail: [nitish.umang@epfl.ch](mailto:nitish.umang@epfl.ch), [nitishumang86@gmail.com](mailto:nitishumang86@gmail.com)

Address : Chemin de la Prairie 62, Apartment 130, 2<sup>nd</sup> floor, 1007 Lausanne, Switzerland

Contact Number: +41 76 7874718, +41 21 6939327

## SELECTED PRESENTATIONS

---

- “Real-Time Management of the Berth Allocation Problem with Stochastic Arrival and Handling Times”, **Umang, N.**, Bierlaire, M. and Erera, A.L. Second INFORMS Transportation Science and Logistics Society Workshop, in Asilomar Conference Grounds, Pacific Grove, California, USA. June 17, 2013.
- “Real-Time Disruption Recovery in Berth Allocation Problem”, **Umang, N.** and Bierlaire, M. 25<sup>th</sup> European Conference on Operational Research (EURO 2012), in Vilnius, Lithuania. July 11, 2012.
- “Integrated and Robust Planning of Bulk Port Operations”, **Umang, N.**, Bierlaire, M. and Robenek, T. Euro Summer Institute on Maritime Logistics (ESI 2012), in Bremen, Germany. June 5, 2012. (*Selected as one of the two EURO/IFORS scholars to attend the workshop*).
- “Disruption Management and Re-scheduling in Berth Allocation Problem in Bulk Ports”, **Umang, N.** and Bierlaire, M. 5<sup>th</sup> International Workshop on Freight Transportation and Logistics (Odysseus 2012), in Mykonos, Greece. May 25, 2012.
- “Exact Solution Approaches to the Berth Allocation Problem in Bulk Sea Port Terminals,” **Umang, N.**, Bierlaire, M., and Vacca, I. 19<sup>th</sup> Triennial Conference of the International Federation of Operational Research Societies (IFORS 2011) in Melbourne, Australia. July 14, 2011.

## TEACHING EXPERIENCE

---

- Spring 2012, EPFL
  - Teaching Assistant (TA) for the Graduate Course *Optimization and Simulation*.
- Spring 2012, Spring 2011, EPFL
  - Teaching Assistant (TA) for the Graduate Course *Decision-aid Methodologies in Transportation*.
- Fall 2013, Fall 2011, EPFL
  - Teaching Assistant (TA) for the Graduate Course *Introduction to Differentiable Optimization*.

## MENTORING EXPERIENCE

---

Master Theses Supervision:

- Integrated berth allocation and yard assignment in bulk ports using column generation, Tomas Robenek (Erasmus Student from DTU, Denmark) – *Best Master Thesis Award for 2013 by the Swiss Operations Research Society*.
- A two-stage sequential linear programming approach to optimize security staff at Geneva Airport, Mahmoud Kharouf (EPFL, Civil Engineering)

Semester Projects Supervision:

- Job Shop Scheduling in a Medical Parts Production Factory, Nathan Scheinmann (EPFL, Mathematics)
- Process Optimization and School Schedules: New Legislation, New Constraints, Zhang Ailin (EPFL, Mathematics)
- Robustness and Recovery in Berth Allocation Problem, Wei Li (EPFL, Management of Technology)

## OTHER PROFESSIONAL ACTIVITIES

---

- Invited referee for major journals including *Transportation Science*, *European Journal of Operational Research*, *Euro Journal on Transportation and Logistics*, *Transportation Research Part B: Methodological*, *Transportation Research Part E: Logistics and Transportation Review*, *Omega*, *Applied Mathematical Modeling* and *Asia-Pacific Journal of Operational Research*.
- Student member of INFORMS.

# Nitish Umang

E-Mail: [nitish.umang@epfl.ch](mailto:nitish.umang@epfl.ch), [nitishumang86@gmail.com](mailto:nitishumang86@gmail.com)

Address : Chemin de la Prairie 62, Apartment 130, 2<sup>nd</sup> floor, 1007 Lausanne, Switzerland

Contact Number: +41 76 7874718, +41 21 6939327

## AWARDS AND HONOURS

---

- Full Scholarship to pursue graduate studies in Civil Engineering (Transportation) at MIT and UC Berkeley, USA, 2008.
- Summer Undergraduate Research Award under the Undergraduate Research Opportunity Program at IIT Delhi, 2006.
- Dean's Merit Scholarship for academic excellence awarded thrice at IIT Delhi.
- Junior Science Talent Search Scholarship by the Government of India, 2001.
- Award for qualifying the Zonal Informatics Olympiad organized by CBSE, 2003.
- Awards for qualifying the Elementary, Junior and Senior Mathematical Olympiads & several other scholarship exams.

## SKILLS

---

- Languages: JAVA, C/C++, MATLAB, SQL, Latex
- Softwares: CPLEX, GLPK, OPL, IDEAS, Fluent, Gambit, ABAQUS, MS Office
- Hobbies: Reading, writing, music and outdoor sports

## PERSONAL INFORMATION

---

- Gender: Male
- Date of Birth: 5<sup>th</sup> October, 1986
- Citizenship: Indian
- Switzerland Residence Permit B

## REFERENCES

---

*Available on request*